



**TUGAS AKHIR - KI141502**

# **3D MULTIPLAYER AR FIGHTING GAME DENGAN KONTROL PERGERAKAN KAMERA**

**ALFIAN MAULANA AZHARI**  
**NRP 5111100704**

**Dosen Pembimbing**  
**Imam Kuswardayan, S.Kom., M.T.**  
**Ridho Rahman Hariadi, S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**  
**SURABAYA**  
**2015**



**FINAL PROJECT- KI141502**

# **3D MULTIPLAYER AR FIGHTING GAME WITH CAMERA MOTION CONTROLLING**

**ALFIAN MAULANA AZHARI  
NRP 5111100704**

**Advisor  
Imam Kuswardayan, S.Kom., M.T.  
Ridho Rahman Hariadi, S.Kom., M.Sc.**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA  
2015**

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji dan syukur kehadiran Allah Subhanahu wa Ta'ala yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “3D MULTIPLAYER AR FIGHTING GAME DENGAN KONTROL PERGERAKAN KAMERA”.

Pengerjaan Tugas Akhir ini adalah momen bagi penulis untuk mengeluarkan seluruh kemampuan, hasrat, dan keinginan penulis sejak pertama kali memasuki dunia kuliah di kampus Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini, penulis mendapatkan banyak bantuan dari berbagai pihak. Melalui lembar ini, penulis ingin menyampaikan ucapan terima kasih secara khusus kepada:

1. Allah SWT atas limpahan rahmat dan rezeki-Nya sehingga penulis dapat menyelesaikan Tugas Akhir.
2. Ayah penulis, Achmad Anwar Ghoni dan Ibu penulis, Nurul Aini, yang selalu memberikan dukungan, doa, perhatian, kasih sayang, serta modal finansial dalam menjalani kuliah.
3. Adik penulis, Ahmad Faisal Hakim, yang telah membantu dalam hal akomodasi dan memberi semangat untuk menyelesaikan studi penulis.
4. Bapak Imam Kuswadaryan selaku dosen pembimbing Tugas Akhir pertama dan yang telah memberikan arahan dalam pengerjaan Tugas Akhir ini.
5. Bapak Ridho Rahman Hariadi selaku dosen pembimbing Tugas Akhir kedua yang dengan sabar membimbing penulis dalam pengerjaan Tugas Akhir ini.
6. Ibu Henning Titi Ciptaningtyas selaku dosen wali yang telah berkenan memberikan saran dan arahan kepada penulis selama menjalani studi.

7. Bapak Radityo Anggoro selaku dosen koordinator Tugas Akhir yang telah membantu penulis atas segala sesuatu yang berkenaan dengan syarat-syarat dan terlaksananya sidang Tugas Akhir.
8. Bapak Darlis Herumurti selaku Ketua Jurusan Teknik Informatika ITS yang selama ini memberikan bantuan kepada penulis baik dalam hal kuliah maupun kegiatan-kegiatan lainnya.
9. Dosen-dosen Teknik Informatika yang dengan sabar mendidik dan memberikan pengalaman baru kepada penulis selama berkuliah di Teknik Informatika.
10. Staf TU Teknik Informatika ITS yang senantiasa memudahkan segala urusan penulis di jurusan.
11. Rekan-rekan dan pengelola Laboratorium Interaksi, Grafika, dan Seni yang telah memberikan fasilitas serta kesempatan melakukan riset atas Tugas Akhir yang dikerjakan penulis.
12. Rekan-rekan dan sahabat-sahabat penulis angkatan 2011 yang telah memberikan dorongan motivasi dan bantuan kepada penulis.
13. Rekan-rekan penulis pada Rumpun Mata Kuliah Interaksi, Grafika, dan Seni yang selalu bahu membahu dalam hal kesuksesan bersama.
14. Pihak-pihak lain yang tidak sengaja terlewat dan tidak dapat penulis sebutkan satu per satu.

Penulis telah berusaha sebaik mungkin dalam menyusun Tugas Akhir ini, namun penulis memohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian dalam pengerjaannya. Kritik dan saran yang membangun dapat disampaikan kepada penulis sebagai bahan perbaikan selanjutnya.

Surabaya, Desember 2015  
Penulis

Alfian Maulana Azhari

## LEMBAR PENGESAHAN

### 3D MULTIPLAYER AR FIGHTING GAME DENGAN KONTROL PERGERAKAN KAMERA

#### Tugas Akhir

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Rumpun Mata Kuliah Interaksi, Grafika, dan Seni  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

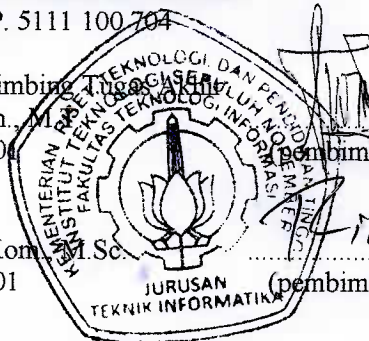
Oleh:

**ALFIAN MAULANA AZHARI**

NRP. 5111 100 704

Disetujui oleh Dosen Pembimbing Tugas Akhir  
Imam Kuswardayan, S.Kom., M. Sc. (pembimbing 1)  
NIP: 19761215 200312 1 001

Ridho Rahman Hariadi, S.Kom., M. Sc. (pembimbing 2)  
NIP: 19870213 201404 1 001



**SURABAYA  
DESEMBER, 2015**

## **3D MULTIPLAYER AR FIGHTING GAME DENGAN KONTROL PERGERAKAN KAMERA**

Nama Mahasiswa : Alfian Maulana Azhari  
NRP : 5111 100 704  
Jurusan : Teknik Informatika FTIf-ITS  
Dosen Pembimbing I : Imam Kuswardayan, S.Kom., M.T.  
Dosen Pembimbing II : Ridho Rahman Hariadi, S.Kom., M.Sc.

### **ABSTRAK**

*Dewasa ini, game merupakan hal yang tidak bisa dipisahkan dari kehidupan manusia, baik yang berusia muda, remaja, maupun tua. Game dapat ditemui dalam banyak platform, seperti desktop maupun mobile. Bahkan, jenisnya pun bermacam-macam.*

*Kebanyakan game selalu mengedepankan unsur adiktif tetapi membuat pemain terlalu santai karena pemain dapat bermain hanya dengan duduk dan memegang perangkat yang bersangkutan sehingga membuat pemain kurang bergerak. Selain itu, kebanyakan game juga hanya menggunakan layar monitor standar sebagai media interaksi dengan pemain sehingga membuat pemain kurang menghiraukan lingkungan sekitar.*

*Dalam Tugas Akhir ini, dibangun suatu game yang ber-genre fighting yang dikombinasikan dengan teknologi Augmented Reality (AR). Game ini dijalankan pada perangkat mobile dengan sistem operasi Android agar dapat dimainkan dengan lebih fleksibel. Tujuan utama dari game ini adalah untuk meningkatkan kemampuan pemain dalam hal berpikir taktis, tingkat interaksi antara pemain, sistem, dan lingkungan sekitar dengan penggunaan marker pada fitur AR, serta gerakan tubuh pemain dengan fitur Motion Gaming.*

***Kata kunci: Unity, Augmented Reality (AR), Perangkat Mobile, Sistem Operasi Android, Fighting Genre, AR Marker, Motion Gaming.***

## **3D MULTIPLAYER AR FIGHTING GAME WITH CAMERA CONTROLLED MOVEMENT**

Student Name : Alfian Maulana Azhari  
NRP : 5111 100 704  
Major : Teknik Informatika FTIf-ITS  
Advisor I : Imam Kuswardayan, S.Kom., M.T.  
Advisor II : Ridho Rahman Hariadi, S.Kom., M.Sc.

### **ABSTRACT**

*Nowadays, game is a thing which cannot be separated from human's life, from the child, teenager, and old. Game can be met in various platforms, such as Desktop and Mobile. Also, there are a lot of genres.*

*Most games always feature addictive material but they make the player's body too relax because the player can play only by sitting down and holding the device so that makes the player move less. Furthermore, most games only use standard monitor screen as the interaction media with the players so that makes the players pay less attention to their surrounding.*

*In this Final Project, a fighting genre game which is combined with the Augmented Reality (AR) will be built. This game runs on mobile device with Android operating system on it so it can be played in a more flexible way. The main goal of this game is to improve the player skill at tactical thinking and the interaction rate between players, system, and the surrounding environments by using marker in the AR feature.*

***Keywords: Unity, Augmented Reality (AR), Mobile Device, Android Operating System, Fighting Genre, AR Marker***



## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK .....	vii
ABSTRACT .....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xix
KODE SUMBER .....	xxi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
1.6 Metodologi .....	3
1.7 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA.....	7
2.1 Unity3D <i>Game Engine</i> .....	7
2.2 <i>Augmented Reality (AR)</i> .....	7
2.3 <i>Photon Unity Networking (PUN)</i> .....	7
2.4 Android.....	8
2.5 <i>Accelerometer</i> .....	9
2.6 Vuforia.....	9
2.7 <i>Fighting Game</i> .....	10
2.8 Blender .....	10
BAB III ANALISIS DAN PERANCANGAN.....	11
3.1 Analisis Sistem .....	11
3.2 Perancangan Permainan.....	12
3.2.1 Deskripsi Umum Perangkat Lunak.....	12
3.2.2 Spesifikasi Kebutuhan Fungsional.....	12
3.2.3 Spesifikasi Kebutuhan Non-Fungsional .....	13
3.2.4 Karakteristik Pengguna.....	13

3.3 Perancangan Sistem.....	14
3.3.1 Perancangan Diagram Kasus Penggunaan.....	14
3.3.2 Perancangan Skenario Kasus Penggunaan.....	15
3.3.3 Perancangan Antarmuka Pengguna .....	18
3.3.4 Perancangan Karakter .....	21
3.3.5 Perancangan Kontrol Permainan.....	48
3.3.6 Perancangan Alur Permainan.....	48
3.3.7 Perancangan Fitur <i>Augmented Reality</i> .....	48
3.3.8 Perancangan Fitur <i>Multiplayer</i> .....	49
BAB IV IMPLEMENTASI .....	51
4.1 Lingkungan Implementasi .....	51
4.2 Implementasi Alur Proses Aplikasi .....	52
4.2.1 Implementasi Main Menu .....	52
4.2.2 Implementasi Character Select Menu .....	54
4.2.3 Implementasi Battle Menu.....	58
4.2.4 Implementasi Kontrol Permainan .....	60
4.2.5 Implementasi <i>Multiplayer</i> .....	67
4.2.6 Implementasi Pertandingan.....	68
BAB V PENGUJIAN DAN EVALUASI .....	73
5.1 Lingkungan Uji Coba .....	73
5.2 Skenario Pengujian Fungsionalitas.....	74
5.2.1 Skenario Pengujian Pemilihan Karakter .....	74
5.2.2 Skenario Pengujian Permainan .....	76
5.3 Pengujian Pengguna.....	82
5.3.1 Skenario Uji Coba Pengguna .....	82
5.3.2 Daftar Penguji Perangkat Lunak.....	83
5.3.3 Hasil Uji Coba Pengguna.....	83
5.3.4 Hasil Pengujian Pengguna .....	86
BAB VI KESIMPULAN DAN SARAN.....	89
6.1. Kesimpulan.....	89
6.2. Saran .....	89
DAFTAR PUSTAKA.....	91
BIODATA PENULIS.....	93

## DAFTAR GAMBAR

Gambar 3.1 Diagram kasus aplikasi.....	15
Gambar 3.2 Diagram aktivitas Memilih Karakter .....	17
Gambar 3.3 Diagram aktivitas Bermain.....	18
Gambar 3.4 Antarmuka Main Menu .....	19
Gambar 3.5 Antarmuka Character Select Menu.....	20
Gambar 3.6 Antarmuka Main Menu .....	21
Gambar 3.7 Tampak kiri karakter Dragon .....	22
Gambar 3.8 Tampak kanan karakter Dragon .....	22
Gambar 3.9 Tampak depan karakter Dragon .....	23
Gambar 3.10 Tampak belakang karakter Dragon.....	23
Gambar 3.11 Tampak atas karakter Dragon.....	24
Gambar 3.12 Tampak bawah karakter Dragon.....	24
Gambar 3.13 Tampak kiri animasi siaga karakter Dragon.....	25
Gambar 3.14 Tampak kanan animasi siaga karakter Dragon.....	25
Gambar 3.15 Tampak kiri animasi berlari karakter Dragon.....	26
Gambar 3.16 Tampak kanan animasi berlari karakter Dragon.....	26
Gambar 3.17 Tampak kiri animasi serangan normal karakter Dragon.....	27
Gambar 3.18 Tampak kanan animasi serangan normal karakter Dragon.....	27
Gambar 3.19 Tampak kiri animasi serangan kuat karakter Dragon .....	28
Gambar 3.20 Tampak kanan animasi serangan kuat karakter Dragon.....	28
Gambar 3.21 Tampak kiri animasi serangan istimewa karakter Dragon.....	29
Gambar 3.22 Tampak kanan animasi serangan istimewa karakter Dragon.....	29
Gambar 3.23 Tampak kiri animasi jatuh karakter Dragon .....	30
Gambar 3.24 Tampak kanan animasi jatuh karakter Dragon .....	30
Gambar 3.25 Tampak atas animasi jatuh karakter Dragon .....	31
Gambar 3.26 Tampak bawah animasi jatuh karakter Dragon .....	31
Gambar 3.27 Tampak kiri karakter Lava Golem.....	32

Gambar 3.28 Tampak kanan karakter Lava Golem.....	32
Gambar 3.29 Tampak depan karakter Lava Golem.....	33
Gambar 3.30 Tampak belakang karakter Lava Golem.....	33
Gambar 3.31 Tampak atas karakter Lava Golem.....	34
Gambar 3.32 Tampak bawah karakter Lava Golem.....	34
Gambar 3.33 Tampak kiri animasi siaga karakter Lava Golem ..	35
Gambar 3.34 Tampak kanan animasi siaga karakter Lava Golem .....	35
Gambar 3.35 Tampak depan animasi siaga karakter Lava Golem .....	36
Gambar 3.36 Tampak belakang animasi siaga karakter Lava Golem.....	36
Gambar 3.37 Tampak kiri animasi berlari karakter Lava Golem	37
Gambar 3.38 Tampak kanan animasi berlari karakter Lava Golem .....	37
Gambar 3.39 Tampak atas animasi berlari karakter Lava Golem	38
Gambar 3.40 Tampak bawah animasi berlari karakter Lava Golem.....	38
Gambar 3.41 Tampak kiri animasi serangan normal karakter Lava Golem.....	39
Gambar 3.42 Tampak kanan animasi serangan normal karkater Lava Golem .....	39
Gambar 3.43 Tampak atas animasi serangan normal karakter Lava Golem .....	40
Gambar 3.44 Tampak bawah animasi serangan normal karakter Lava Golem .....	40
Gambar 3.45 Tampak kiri animasi serangan kuat karakter Lava Golem.....	41
Gambar 3.46 Tampak kanan animasi serangan kuat karakter Lava Golem .....	41
Gambar 3.47 Tampak depan animasi serangan kuat karakter Lava Golem.....	42
Gambar 3.48 Tampak belakang animasi serangan kuat karakter Lava Golem .....	42

Gambar 3.49 Tampak kiri animasi serangan istimewa karakter Lava Golem .....	43
Gambar 3.50 Tampak kanan animasi serangan istimewa karakter Lava Golem .....	44
Gambar 3.51 Tampak depan animasi serangan istimewa karakter Lava Golem .....	44
Gambar 3.52 Tampak belakang animasi serangan istimewa karakter Lava Golem .....	45
Gambar 3.53 Tampak kiri animasi jatuh karakter Lava Golem ..	46
Gambar 3.54 Tampak kiri animasi jatuh karakter Lava Golem ..	46
Gambar 3.55 Tampak atas animasi jatuh karakter Lava Golem ..	47
Gambar 3.56 Tampak bawah animasi jatuh karakter Lava Golem .....	47
Gambar 3.57 <i>Marker</i> untuk fitur AR .....	49

## DAFTAR TABEL

Tabel 3.1 Karakteristik Pengguna .....	14
Tabel 3.2 Skenario Kasus Penggunaan .....	15
Tabel 3.3 Skenario Kasus Penggunaan Memilih Karakter .....	16
Tabel 3.4 Skenario Kasus Penggunaan Bermain .....	16
Tabel 3.5 Status <i>damage</i> serangan dari masing-masing karakter .....	21
Tabel 4.1 Lingkungan Implementasi Perangkat Lunak (1) .....	51
Tabel 4.2 Lingkungan Implementasi Perangkat Lunak (2) .....	51
Tabel 4.3 Lingkungan Implementasi Perangkat Lunak (3) .....	51
Tabel 5.1 Lingkungan Uji Coba Perangkat Lunak (1) .....	73
Tabel 5.2 Lingkungan Uji Coba Perangkat Lunak (2) .....	73
Tabel 5.3 Lingkungan Uji Coba Perangkat Lunak (3) .....	74
Tabel 5.4 Pengujian Pemilihan Karakter .....	74
Tabel 5.5 Pengujian Permainan .....	76
Tabel 5.6 Rekapitulasi hasil pengujian fungsional .....	82
Tabel 5.7 Daftar Nama Penguji Coba Aplikasi .....	83
Tabel 5.8 Skala dan kategori penilaian .....	84
Tabel 5.9 Penilaian Antarmuka .....	84
Tabel 5.10 Penilaian Performa Sistem .....	85
Tabel 5.11 Rekapitulasi Hasil Uji Coba Pengguna .....	86

## KODE SUMBER

Kode Sumber 4.1 Tombol “PLAY” .....	53
Kode Sumber 4.2 Tombol “EXIT” .....	53
Kode Sumber 4.3 Pemilihan karakter .....	56
Kode Sumber 4.4 Tombol karakter .....	56
Kode Sumber 4.5 Tombol “FIGHT!” .....	57
Kode Sumber 4.6 Tombol “Esc” .....	58
Kode Sumber 4.7 Tombol maju .....	61
Kode Sumber 4.8 Tombol mundur .....	62
Kode Sumber 4.9 Perintah rotasi dengan <i>accelerometer</i> .....	63
Kode Sumber 4.10 Tombol “Attack” .....	64
Kode Sumber 4.11 Tombol “Power Attack” .....	65
Kode Sumber 4.12 Tombol “Special Attack” .....	66
Kode Sumber 4.13 Tombol “Esc” pada antarmuka Character Select Menu .....	67
Kode Sumber 4.14 Pembuatan <i>room</i> pada <i>server</i> Photon .....	68
Kode Sumber 4.15 Instansiasi karakter pada arena .....	68
Kode Sumber 4.16 Status HP dan MP <i>default</i> karakter .....	69
Kode Sumber 4.17 Pengaturan perubahan status HP dan MP .....	70
Kode Sumber 4.18 Menampilkan status HP dan MP .....	70
Kode Sumber 4.19 Notifikasi ( <i>popup</i> ) dan pengembalian pemain ke antarmuka Character Select .....	71

# **BAB I**

## **PENDAHULUAN**

Bab ini memaparkan garis besar Tugas Akhir yang meliputi latar belakang, tujuan dan manfaat pembuatan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

### **1.1 Latar Belakang**

Kemajuan teknologi informasi menjadikan dunia *gaming* semakin bermacam-macam jenisnya baik dalam hal *genre*, *gameplay*, fitur, serta jenis grafis yang diimplementasikan seperti 2D dan 3D. Di pasaran, terdapat banyak *game* yang mengimplementasikan grafis 2D yang dikombinasikan dengan 3D. Tetapi, saat ini, kebanyakan *game* yang mengimplementasikan grafis 3D hanya sebatas implementasi pada layar monitor. Implementasi grafis 3D yang berhubungan langsung dengan objek nyata tiga dimensi masih jarang. Salah satu contoh implementasi grafis 3D seperti di atas adalah *Augmented Reality* (AR) atau Realitas Tertambah.

Teknologi AR adalah sebuah teknologi baru yang melibatkan lapisan grafis komputer di atas objek dunia nyata. Teknologi *Augmented Reality* baru berkembang dan dapat dinikmati khalayak ramai pada awal tahun 2000-an. Tetapi, masih banyak orang yang belum mengenal teknologi ini dikarenakan minimnya publikasi, kurangnya perangkat yang diperlukan, serta biaya yang relatif mahal. Teknologi ini menggunakan kamera sebagai masukan kepada sistem dan sebagai alat untuk membaca *marker*. Pada awal perkembangannya, teknologi ini menggunakan perangkat komputer biasa sehingga pengguna membutuhkan biaya tambahan untuk kamera. Dengan perangkat Android, pengguna tidak membutuhkan biaya tambahan tersebut karena sebagian besar perangkat Android telah memiliki kamera.

Nama *game* ini adalah *Creature Drive*. Nama ini diambil karena pemain seolah-olah sedang mengendarai seekor makhluk atau



monster dengan cara yang mirip dengan mengendarai mobil atau kendaraan sejenisnya. *Game* ini akan berjalan pada *platform* Android dan memiliki fitur AR serta *multiplayer* antar 2 pemain.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana membuat *game* ber-genre *fighting* berbasis 3D pada *platform* Android yang mengakomodasi AR?
2. Bagaimana mengimplementasikan Photon Unity Networking pada *platform* Android untuk berkomunikasi dengan sesama perangkat Android?
3. Bagaimana menentukan kontrol gerakan karakter dengan *Accelerometer*?

## 1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. *Game* ini berbasis *mobile* pada *platform* Android
2. Aplikasi dibangun dengan menggunakan *Game Engine* Unity3D dan bahasa pemrograman C#
3. Jumlah pemain maksimal adalah 2 orang
4. Perangkat yang digunakan oleh pemain harus tersambung dengan internet (tanpa *proxy*)

## 1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini antara lain:

1. Membangun suatu sistem *game* yang mengimplementasikan AR dan *multiplayer*
2. Mempublikasikan teknologi AR kepada khalayak ramai
3. Meningkatkan interaksi sosial antar pengguna
4. Memicu aktivitas tubuh pengguna

## 1.5 Manfaat

Manfaat dari hasil pembuatan Tugas Akhir ini antara lain:

1. Sebagai bentuk implementasi teknologi AR
2. Menghibur pengguna dengan *game* yang interaktif
3. Meningkatkan pola pikir strategis

## 1.6 Metodologi

Pembuatan Tugas Akhir dilakukan menggunakan metodologi sebagai berikut:

### A. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu sebagai berikut:

1. Unity3D;
2. *Augmented Reality* (AR);
3. Photon Unity Networking;
4. Bahasa Pemrograman C#;
5. *Accelerometer*.

### B. Perancangan perangkat lunak

Pada tahap ini, dilakukan analisa awal dan pendefinisian kebutuhan sistem untuk mengetahui permasalahan yang sedang dihadapi. Selanjutnya, dirumuskan rancangan sistem yang dapat memberi solusi terhadap permasalahan tersebut. Langkah yang akan digunakan pada tahap ini adalah sebagai berikut:

1. Pencarian dan pendataan materi yang akan digunakan dalam *game Creature Drive*
2. Perancangan sistem dan mekanisme *game Creature Drive*
3. Analisis kebutuhan non fungsional.
4. Analisis algoritma dan formula dalam membangun *game Creature Drive*

### C. Implementasi dan pembuatan sistem

*Game* ini akan dibangun dengan bahasa pemrograman C# dan *Game Engine* Unity3D. Untuk mendesain karakter, *modeling tools* yang akan digunakan adalah Blender.

#### D. Uji coba dan evaluasi

Pada tahap ini, akan dilakukan pengujian terhadap perangkat lunak menggunakan data atau skenario yang telah dipersiapkan sebelumnya yakni sebagai berikut:

##### 1. Pengujian *black-box*

Pengujian *black-box* adalah pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetestan pada spesifikasi fungsional program. Pengujian ini dilakukan untuk menguji apakah proses kinerja aplikasi *game* ini sudah sesuai dengan kebutuhan pengguna atau tidak [1].

##### 2. Pengujian usabilitas

Pengujian usabilitas dilakukan dengan cara melakukan survei ke pengguna Android di sekitar lingkungan Teknik Informatika ITS. Survei dilakukan untuk mengukur tingkat kepuasan dalam hal hiburan dari *game* yang dibuat.

#### E. Penyusunan laporan tugas akhir

Pada tahap ini, dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam Tugas Akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

### 1.7 Sistematika Penulisan

Buku Tugas Akhir ini terdiri dari beberapa bab, antara lain sebagai berikut.

## **BAB I PENDAHULUAN**

Bab ini berisi latar belakang masalah, rumusan dan batasan permasalahan, tujuan dan manfaat pembuatan tugas akhir, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

## **BAB II TINJAUAN PUSTAKA**

Bab ini membahas dasar pembuatan dan beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang mendasari pembuatan tugas akhir ini.

### **BAB III ANALISIS DAN PERANCANGAN**

Bab ini membahas analisis dari sistem yang dibuat meliputi analisis permasalahan, deskripsi umum perangkat lunak, spesifikasi kebutuhan, dan identifikasi pengguna. Kemudian membahas rancangan dari sistem yang dibuat meliputi rancangan skenario kasus penggunaan, arsitektur, data, dan antarmuka.

### **BAB IV IMPLEMENTASI**

Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembangunan area permainan, dan antarmuka permainan.

### **BAB V PENGUJIAN DAN EVALUASI**

Bab ini membahas pengujian dari aplikasi yang dibuat dengan melihat keluaran yang dihasilkan oleh aplikasi dan evaluasi untuk mengetahui kemampuan aplikasi.

### **BAB VI PENUTUP**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan serta saran untuk pengembangan aplikasi selanjutnya.

## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir ini. Teori-teori tersebut *adalah Unity3D Game Engine, Augmented Reality (AR), Photon Unity Networking (PUN), Android, Fighting Game, dan Rancang Bangun Perangkat Lunak.*

#### **2.1 Unity3D Game Engine**

Unity3D adalah sebuah *game engine* yang dikembangkan oleh Unity Technologies yang juga merupakan *tool* data yang digunakan tidak hanya untuk membuat *game* tetapi juga konten, animasi 3D waktu nyata, dll. Unity dapat berjalan pada Windows dan Macintosh OS X dan digunakan untuk pengembangan pada *platform* yang bermacam-macam termasuk Windows, Mac, Will, iPhone, Android, dll. Unity juga dapat digunakan untuk membuat *game* 3D berbasis *web* dengan menggunakan *plug-in* Unity *Web Player* [2].

#### **2.2 Augmented Reality (AR)**

*Augmented Reality (AR)* adalah sebuah teknologi baru untuk menambah dunia nyata dengan melapisi data grafis dari komputer di atasnya. Karakteristik dari AR ada 3 hal, yaitu: (1) mencampur gambar virtual dengan dunia nyata, (2) registrasi data digital 3D, dan (3) interaktivitas waktu nyata. AR menggunakan *marker* sebagai penanda dimana objek virtual yang dimaksud akan dilapiskan pada dunia nyata [3].

#### **2.3 Photon Unity Networking (PUN)**

Photon adalah *framework* pengembangan *game* berjenis *real-time multiplayer* yang cepat, ringan, dan fleksibel. Photon terdiri dari sebuah *server* dan beberapa SDK *client* untuk banyak *platform*. Photon Unity Networking (PUN) adalah *framework* istimewa pada *client* yang ditujukan untuk mengimplementasikan ulang dan

menambah fitur pada jaringan Unity. Unity menggunakan fitur Photon untuk mengomunikasikan dan memasang para pemain [4].

## 2.4 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Antarmuka pengguna Android didasarkan pada manipulasi langsung, menggunakan masukan sentuh yang serupa dengan tindakan di dunia nyata, seperti menggesek, mengetuk, mencubit, dan membalikkan cubitan untuk memanipulasi obyek di layar. Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (*apps*) yang memperluas fungsionalitas perangkat dan umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2012, ada sekitar 700.000 aplikasi yang tersedia untuk Android, dan sekitar 25 juta aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android adalah *platform* paling populer bagi para pengembang yang digunakan oleh 71% pengembang aplikasi seluler [5].

## 2.5 *Accelerometer*

*Accelerometer* adalah sebuah perangkat yang mampu mengukur sebuah kekuatan akselerasi. Kekuatan ini mungkin statis (diam) seperti halnya kekuatan konstan dari gravitasi Bumi, atau bisa juga bersifat dinamis karena gerakan atau getaran dari sebuah alat akselerometer.

*Accelerometer* adalah sebuah transduser yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan akibat gravitasi bumi. *Accelerometer* juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan, bangunan, mesin, juga bisa digunakan untuk mengukur getaran yang terjadi di dalam bumi, getaran mesin, jarak yang dinamis, dan kecepatan dengan ataupun tanpa pengaruh gravitasi bumi [6].

## 2.6 *Vuforia*

Vuforia merupakan perangkat lunak untuk *Augmented Reality* yang menggunakan sumber yang konsisten mengenai *computer vision* yang fokus pada *image recognition*. Vuforia mempunyai banyak fitur dan kemampuan yang dapat membantu pengembang untuk mewujudkan pemikiran mereka tanpa adanya batas secara teknis. Dengan dukungan untuk iOS, Android, dan Unity3D, *platform* Vuforia mendukung para pengembang untuk membuat aplikasi yang dapat digunakan di hampir seluruh jenis *smartphone* dan *tablet*. Pengembang juga diberikan kebebasan untuk mendesain dan membuat aplikasi yang mempunyai kemampuan antara lain :

1. Teknologi *computer vision* tingkat tinggi yang memungkinkan pengembang untuk membuat efek khusus pada perangkat *mobile*.
2. Terus-menerus mengenali *multiple image*.
3. *Tracking* dan *Detection* tingkat lanjut.
4. Solusi pengaturan basis data gambar yang fleksibel.

Target pada vuforia merupakan obyek pada dunia nyata yang dapat dideteksi oleh kamera untuk menampilkan obyek virtual. Beberapa jenis target pada vuforia adalah :

1. *Image target*, contoh : foto, papan permainan, halaman majalah, sampul buku, kemasan produk, poster, kartu ucapan, dll. Jenis target ini menampilkan gambar sederhana dari *Augmented Reality*.
2. *Frame marker*, tipe *frame* gambar 2D dengan pola khusus yang dapat digunakan sebagai potongan permainan di permainan pada papan.
3. *Multi-target*, misalnya kemasan produk atau produk yang berbentuk kotak ataupun persegi. Jenis ini dapat menampilkan gambar sederhana *Augmented Reality* 3D.
4. *Virtual button*, yang dapat membuat tombol pada suatu daerah kotak pada *image target*. Pada Vuforia, ada 2 jenis *workflow* dengan dasar basis data yang dapat dipilih oleh pengembang, yaitu *Cloud Database* dan *Device Database* [7].

## 2.7 *Fighting Game*

*Fighting Game* adalah salah satu *genre game* dimana seorang pemain mengontrol karakter dan melakukan pertarungan jarak dekat dengan lawan. Karakter-karakter dari *game ber-genre* ini cenderung seimbang dan bertarung dalam suatu pertandingan dengan beberapa ronde pada sebuah arena. Pemain harus menguasai teknik seperti menahan, membalas serangan, dan menyambungkan serangan yang sering disebut dengan *combo*. Sejak awal tahun 1990-an, sebagian besar *fighting game* menjadikan pemain dapat melakukan serangan istimewa dengan menggunakan kombinasi tombol tertentu [8].

## 2.8 *Blender*

Blender adalah perangkat pembuatan konten 3D yang terintegrasi penuh yang menawarkan bermacam-macam perangkat penting, termasuk pemodelan, *UV Mapping*, *Texturing*, *Rigging*, animasi, berbagai macam simulasi, *Scripting*, *Rendering*, *Compositing*, VFX, dan pembuatan permainan [9].



## BAB III ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas analisis dan perancangan yang akan digunakan untuk menyelesaikan Tugas Akhir ini. Dalam Tugas Akhir ini akan dibuat sebuah *game* dengan jenis *Fighting* yang berbasis 3D.

### 3.1 Analisis Sistem

Kemajuan teknologi informasi menjadikan dunia *gaming* semakin bermacam-macam jenisnya baik dalam hal *genre*, *gameplay*, maupun fitur, serta jenis grafis yang diimplementasikan seperti 2D dan 3D. Di pasar, terdapat banyak *game* yang mengimplementasikan grafis 2D yang dikombinasikan dengan 3D. Tetapi, saat ini, kebanyakan *game* yang mengimplementasikan grafis 3D hanya sebatas implementasi pada layar monitor. Implementasi grafis 3D yang berhubungan langsung dengan objek nyata tiga dimensi masih jarang. Salah satu contoh implementasi grafis 3D seperti di atas adalah *Augmented Reality* (AR) atau Realitas Tertambah.

Aplikasi ini dibangun dengan tujuan memberikan hiburan kepada pengguna dengan *game* yang interaktif dan inovatif. *Game* ini juga diharapkan dapat meningkatkan pola pikir taktis pemain dengan fitur *fighting* dan untuk meningkatkan gerakan tubuh pemain ketika bermain *game* dengan fitur *Motion Gaming*.

Penulis menggunakan *Game Engine* Unity3D dengan bahasa pemrograman C# untuk memfasilitasi pengembangan aplikasi ini. Penulis menggunakan *package* atau *library* Vuforia AR untuk mengakomodasi *Augmented Reality* (AR), *package* atau *library* Photon Unity Network (PUN) untuk mengakomodasi fitur *multiplayer*, serta *accelerometer* untuk mengontrol karakter.

## 3.2 Perancangan Permainan

### 3.2.1 Deskripsi Umum Perangkat Lunak

Perangkat lunak yang dikembangkan adalah sebuah permainan berbasis 3D berjenis *fighting*. Permainan ini mengakomodasi teknologi AR dan pengontrolan gerakan karakter menggunakan *accelerometer* sehingga terasa lebih nyata. Permainan ini ditujukan untuk dimainkan oleh 2 orang. Setiap karakter yang dipilih pemain memiliki status *Health Point* (HP) dan *Mental Point* (MP). Status HP menunjukkan sisa nyawa dari suatu karakter sehingga KO atau kalah. Status MP menunjukkan bisa atau tidaknya suatu karakter menggunakan *Power Attack* atau *Special Attack*-nya. Nilai MP akan bertambah apabila serangan normal suatu karakter mengenai karakter lawan atau karakter tersebut terkena serangan dari karakter lawan. Nilai MP akan berkurang apabila suatu karakter menggunakan *Power Attack* atau *Special Attack*-nya. Permainan ini dijalankan pada perangkat Android.

Untuk mulai bermain, salah satu pemain harus memiliki *marker* khusus yang telah ditentukan. *Marker* tersebut sebaiknya dicetak terlebih dahulu. Setelah membuka aplikasi, kedua pemain harus menekan tombol “Play” pada antarmuka **Main Menu**. Kemudian, kedua pemain memilih karakter yang dikehendaki pada antarmuka **Character Select Menu**. Setelah itu, karakter yang dipilih oleh kedua pemain akan memasuki arena permainan pada antarmuka **Battle Menu**. Pemain yang karakternya berhasil menghabiskan HP lawan terlebih dahulu dinyatakan sebagai pemenang dan pemain yang HP karakternya habis terlebih dahulu dinyatakan kalah. Setelah salah satu pemain dinyatakan menang, kedua pemain akan dikembalikan ke **Menu Character Select** sehingga mereka dapat memilih karakter dan bermain lagi.

### 3.2.2 Spesifikasi Kebutuhan Fungsional

Berdasarkan deskripsi umum sistem, dapat disimpulkan bahwa kebutuhan fungsional dari aplikasi ini adalah dapat digunakan untuk bermain *game*.

### 3.2.3 Spesifikasi Kebutuhan Non-Fungsional

Terdapat beberapa kebutuhan non-fungsional yang apabila dipenuhi, dapat meningkatkan kualitas dari permainan ini. Berikut daftar kebutuhan non-fungsional:

#### 1. *Frame Rate*

*Game* ini harus dapat dimainkan dengan lancar dan tidak ada *lag*. *Game* 3D biasanya akan menjadi optimal dengan *Frame Rate* di atas 30 fps (*Frame per Second*). Namun, instansiasi karakter pada arena dapat mempengaruhi laju *Frame Rate*. Laju *Frame Rate* juga dipengaruhi oleh spesifikasi dari perangkat Android yang bersangkutan.

#### 2. *Grafis*

Kenyamanan bermain berbanding lurus dengan kualitas grafis yang disajikan dalam permainan. Efek seperti animasi merupakan salah satu daya tarik dalam suatu permainan. Efek-efek ini bisa membuat *rate fps* menjadi berkurang dan permainan melambat (*lag*) karena membutuhkan tambahan komputasi.

#### 3. *Koneksi Internet*

Permainan ini mengakomodasi *multiplayer* yang menggunakan internet sebagai media penghubung antar pemain sehingga koneksi internet yang lancar dibutuhkan supaya permainan tidak mengalami *lag*.

### 3.2.4 Karakteristik Pengguna

Berdasarkan deskripsi umum di atas, maka dapat diketahui bahwa pengguna yang akan menggunakan aplikasi ini ada dua jenis, yaitu pemain dan pengembang. Karakteristik pengguna tercantum dalam Tabel 3.1.

**Tabel 3.1 Karakteristik Pengguna**

<b>Nama Aktor</b>	<b>Tugas</b>	<b>Hak Akses Aplikasi</b>	<b>Kemampuan yang harus dimiliki</b>
Pemain	Pihak yang menggunakan	Menggunakan aplikasi	Tidak ada
Pengembang	Pihak yang mengembangkan	Membuat dan mengubah aplikasi	Pemrograman dan pengembangan <i>game</i>

### **3.3 Perancangan Sistem**

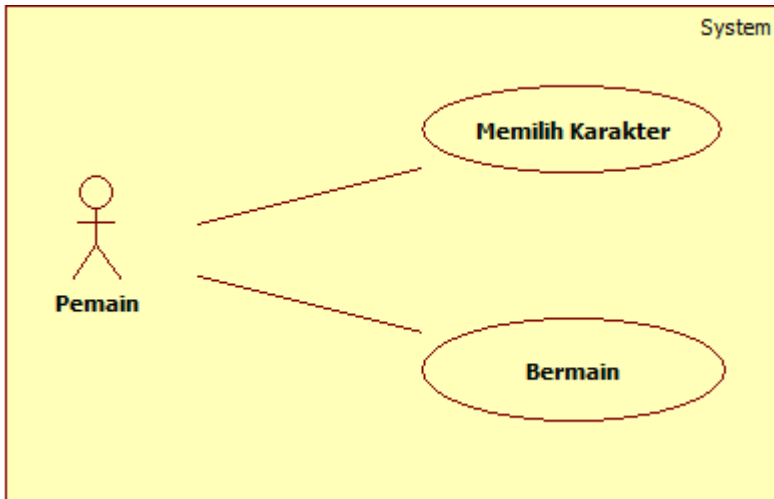
Tahap perancangan dalam subbab ini dibagi menjadi beberapa bagian yaitu perancangan diagram kasus penggunaan, perancangan skenario kasus penggunaan, perancangan antarmuka, perancangan kontrol permainan, dan perancangan diagram alur permainan.

#### **3.3.1 Perancangan Diagram Kasus Penggunaan**

Dalam aplikasi Tugas Akhir ini, terdapat dua kasus penggunaan, antara lain memilih karakter dan bermain. Pengguna atau entitas luar dari sistem adalah pemain.

### 3.3.2 Perancangan Skenario Kasus Penggunaan

Kasus penggunaan yang terdapat di dalam sistem ditunjukkan pada Gambar 3.1.



**Gambar 3.1 Diagram kasus aplikasi**

Penjelasan dari masing-masing kasus penggunaan dicantumkan pada Tabel 3.2. Tabel tersebut berisi penjelasan skenario yang akan dilakukan ketika pengujian.

**Tabel 3.2 Skenario Kasus Penggunaan**

No	Kode Kasus Penggunaan	Nama Kasus Penggunaan	Keterangan
1	UC-001	Memilih karakter	Pemain memilih karakter yang dikehendaki untuk dimainkan
2	UC-002	Bermain	Pemain memainkan karakter yang telah dipilih

### 3.3.2.1 Kasus Penggunaan Permainan

Penjelasan kasus penggunaan permainan untuk skenario UC-001 yakni Memilih karakter ditunjukkan pada Tabel 3.3.

**Tabel 3.3 Skenario Kasus Penggunaan Memilih Karakter**

<b>Nama Kasus Penggunaan</b>	Memilih karakter
<b>Kode</b>	UC-001
<b>Deskripsi</b>	Kasus penggunaan dimana aktor memilih karakter yang hendak dimainkan
<b>Aktor</b>	Pemain
<b>Kondisi Awal</b>	<i>Marker</i> khusus sudah tersedia, pemain sudah masuk ke aplikasi dan sistem memunculkan antarmuka <b>Main Menu</b> .
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pemain menekan tombol “Play”.</li> <li>2. Sistem menampilkan antarmuka <b>Character Select Menu</b>.</li> <li>3. Pemain memilih karakter.</li> <li>4. Pemain menekan tombol “FIGHT!”.</li> </ol>

Selanjutnya, penjelasan kasus penggunaan permainan untuk skenario UC-002 yakni Bermain ditunjukkan pada Tabel 3.4.

**Tabel 3.4 Skenario Kasus Penggunaan Bermain**

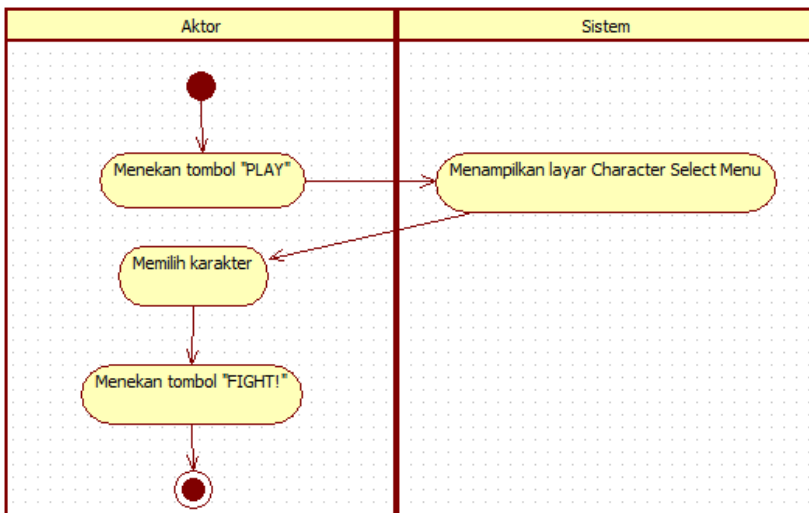
<b>Nama Kasus Penggunaan</b>	Bermain
<b>Kode</b>	UC-002
<b>Deskripsi</b>	Kasus penggunaan dimana aktor bermain dengan karakter yang telah dipilih.
<b>Aktor</b>	Pemain
<b>Kondisi Awal</b>	Pemain sudah masuk ke aplikasi dan sistem memunculkan antarmuka <b>Character Select Menu</b> .
<b>Alur Normal</b>	<ol style="list-style-type: none"> <li>1. Pemain menekan tombol “FIGHT!” pada <b>Character Select Menu</b>.</li> <li>2. Sistem menampilkan antarmuka <b>Battle Menu</b>.</li> </ol>

- |  |   |
|--|---|
|  | 3. Permainan selesai.<br>4. Sistem menampilkan antarmuka <b>Character Select Menu</b> . |
|--|---|

### 3.3.2.2 Diagram Aktivitas

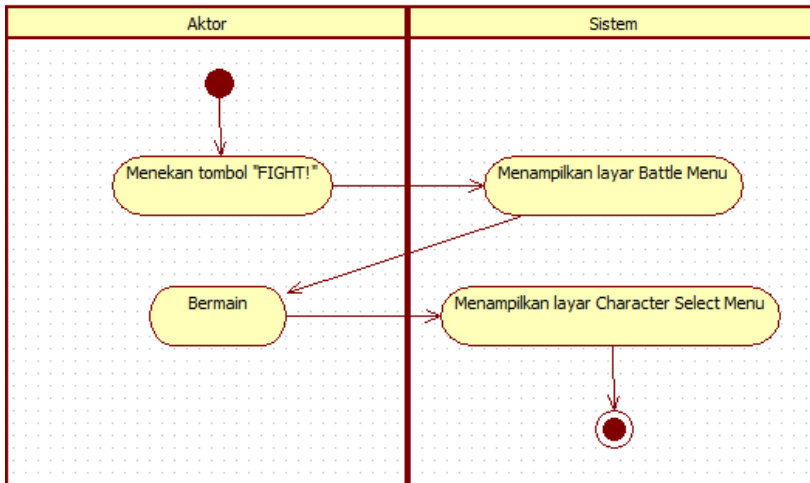
Diagram aktivitas menampilkan langkah-langkah normal yang harus dilakukan pemain untuk menjalankan studi kasus permainan dimulai dari awal permainan hingga kondisi akhir.

Diagram aktivitas dari kasus penggunaan UC-001 yakni Memilih karakter ditunjukkan pada Gambar 3.2.



**Gambar 3.2 Diagram aktivitas Memilih Karakter**

Kemudian, diagram aktivitas dari kasus penggunaan UC-002 yakni Bermain ditunjukkan pada Gambar 3.3.



**Gambar 3.3 Diagram aktivitas Bermain**

### 3.3.3 Perancangan Antarmuka Pengguna

Subbab ini membahas rancangan antarmuka pengguna yang akan digunakan pada Tugas Akhir ini. Rancangan antarmuka yang dibahas meliputi ketentuan masukan dan rancangan jendela antarmuka. Dalam aplikasi ini terdapat beberapa antarmuka, yaitu **Main Menu**, **Character Select Menu**, dan **Battle Menu**.

#### 3.3.3.1 Antarmuka Main Menu

Antarmuka Main Menu merupakan antarmuka yang pertama kali muncul ketika aplikasi dijalankan. Antarmuka ini berisi 2 tombol, yaitu tombol “Play” untuk menuju ke antarmuka **Character Select Menu** dan “Exit” untuk keluar dari aplikasi. Antarmuka **Main Menu** ditunjukkan pada Gambar 3.4.

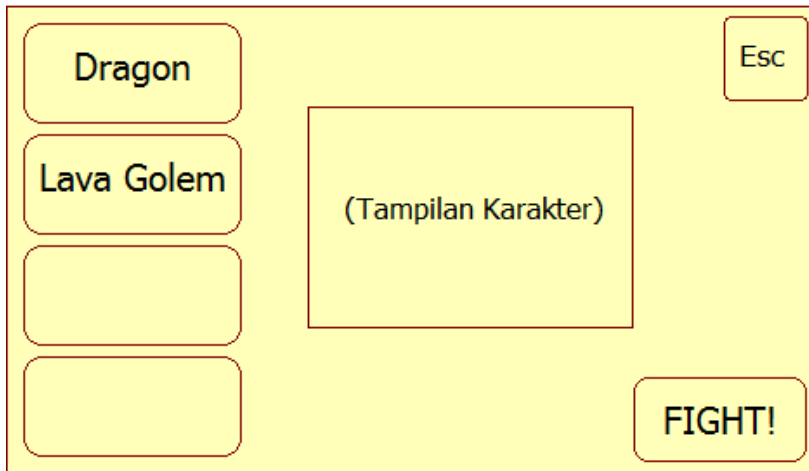




**Gambar 3.4 Antarmuka Main Menu**

### **3.3.3.2 Antarmuka Character Select Menu**

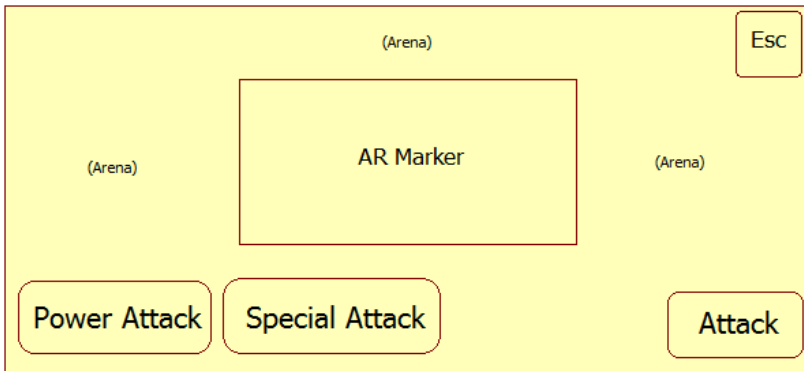
Tampilan **Character Select Menu** adalah antarmuka yang tampil setelah tombol “Play” pada tampilan **Main Menu** ditekan. Antarmuka ini digunakan untuk memilih karakter yang akan digunakan untuk bermain sesuai dengan keinginan pemain. Setelah pemain memastikan karakter yang ingin dimainkan, pemain mengeklik tombol “FIGHT!” untuk masuk ke Tampilan **Battle Menu**. Empat tombol di sebelah kiri digunakan untuk memilih karakter. Tombol “Esc” di kanan atas digunakan untuk kembali ke tampilan Main Menu. Tampilan **Character Select Menu** ditunjukkan pada Gambar 3.5.



**Gambar 3.5 Antarmuka Character Select Menu**

### **3.3.3.3 Antarmuka Battle Menu**

Antarmuka **Battle Menu** merupakan antarmuka yang tampil setelah tombol “FIGHT!” pada Antarmuka **Character Select Menu** ditekan. Di tengah arena terdapat *marker* untuk fitur AR dan di sebelah kanan atas terdapat tombol “Esc”. Ada 3 tombol untuk menyerang yaitu “Attack”, “Power Attack”, dan “Special Attack”. Antarmuka **Battle Menu** ditunjukkan pada Gambar 3.6.



**Gambar 3.6 Antarmuka Main Menu**

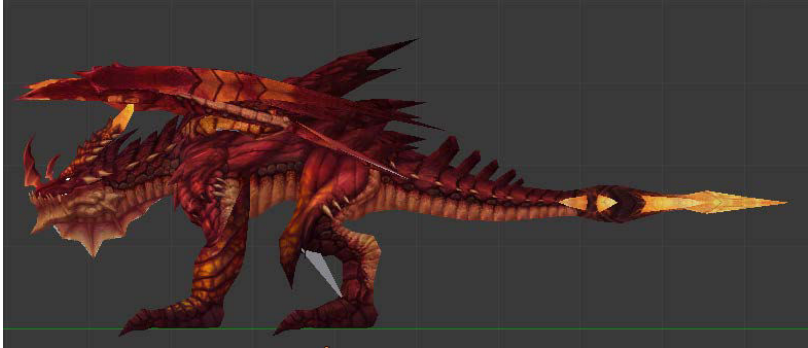
### 3.3.4 Perancangan Karakter

Subbab ini membahas tentang rancangan karakter yang akan digunakan dalam *game*. Karakter yang akan digunakan berbasis 3D. Dalam *game* ini, terdapat 2 macam karakter yaitu Dragon dan Lava Golem. Karakter-karakter tersebut dibuat dengan menggunakan aplikasi pemodelan Blender. Setiap karakter memiliki bentuk, animasi, dan status yang berbeda. Status-status tersebut meliputi *Health Point* (HP), *Mental Point* (MP), kerusakan serangan normal (*Normal Attack Damage*), kerusakan serangan kuat (*Power Attack Damage*), kerusakan serangan istimewa (*Special Attack Damage*), dan kecepatan gerak. *Damage* dari setiap serangan dari setiap karakter ditunjukkan pada Tabel 3.5.

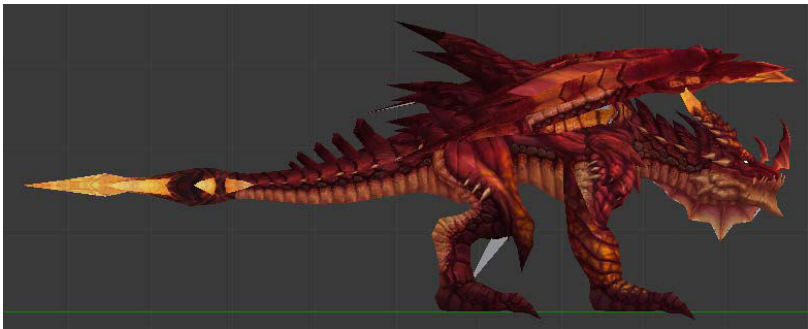
**Tabel 3.5 Status *damage* serangan dari masing-masing karakter**

Nama Karakter	Damage		
	<i>Normal Attack</i>	<i>Power Attack</i>	<i>Special Attack</i>
Dragon	20	150	400
Lava Golem	25	165	450

Tampilan karakter Dragon dari beberapa sudut pandang ditunjukkan pada Gambar 3.7, Gambar 3.8, Gambar 3.9, Gambar 3.10, Gambar 3.11, dan Gambar 3.12.



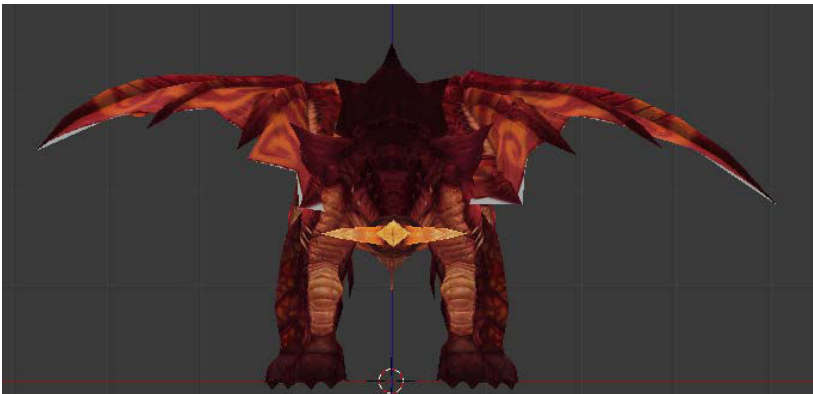
**Gambar 3.7 Tampak kiri karakter Dragon**



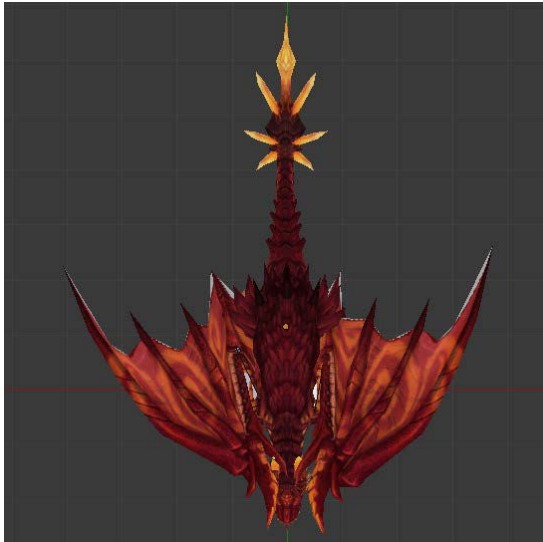
**Gambar 3.8 Tampak kanan karakter Dragon**



**Gambar 3.9 Tampak depan karakter Dragon**



**Gambar 3.10 Tampak belakang karakter Dragon**

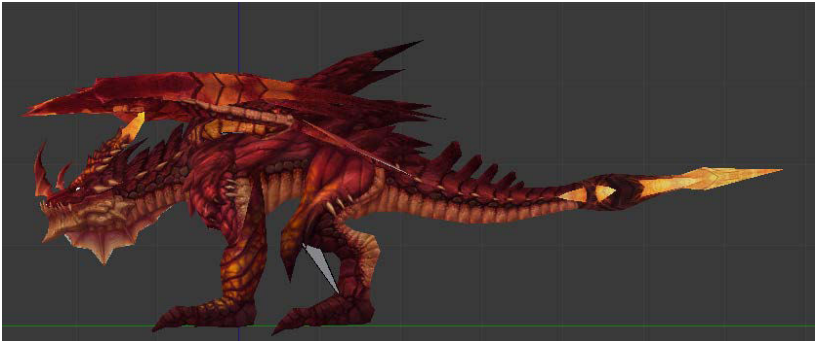


**Gambar 3.11 Tampak atas karakter Dragon**

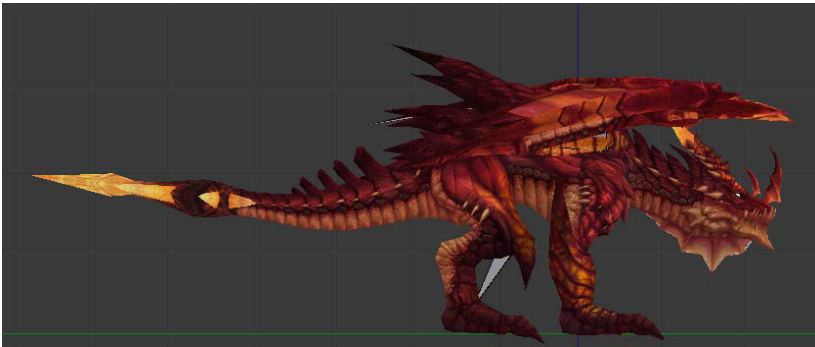


**Gambar 3.12 Tampak bawah karakter Dragon**

Tampilan animasi siaga (*idle*) karakter Dragon dari beberapa sudut pandang ditunjukkan pada Gambar 3.13 dan Gambar 3.14. Animasi ini dijalankan sebagai animasi asal atau *default*.



**Gambar 3.13 Tampak kiri animasi siaga karakter Dragon**

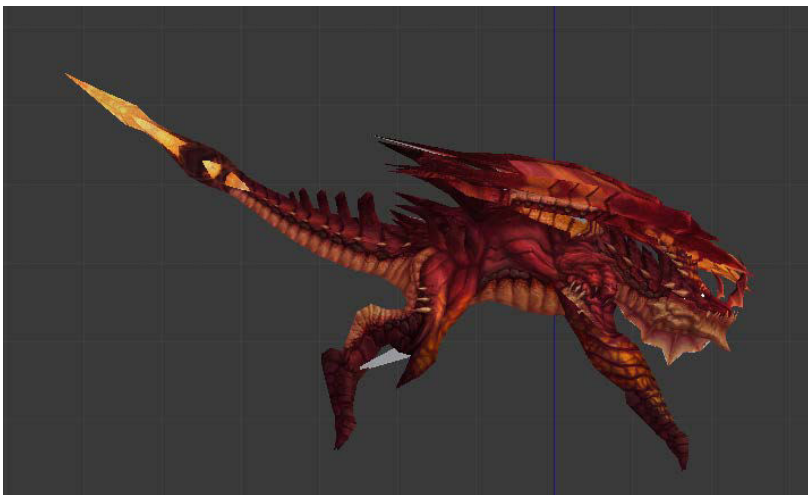


**Gambar 3.14 Tampak kanan animasi siaga karakter Dragon**

Tampilan animasi berlari karakter Dragon dari beberapa sudut pandang ditunjukkan pada Gambar 3.15 dan Gambar 3.16. Animasi ini dijalankan ketika tombol maju atau mundur ditekan dan karakter bertranslasi.



**Gambar 3.15 Tampak kiri animasi berlari karakter Dragon**



**Gambar 3.16 Tampak kanan animasi berlari karakter Dragon**

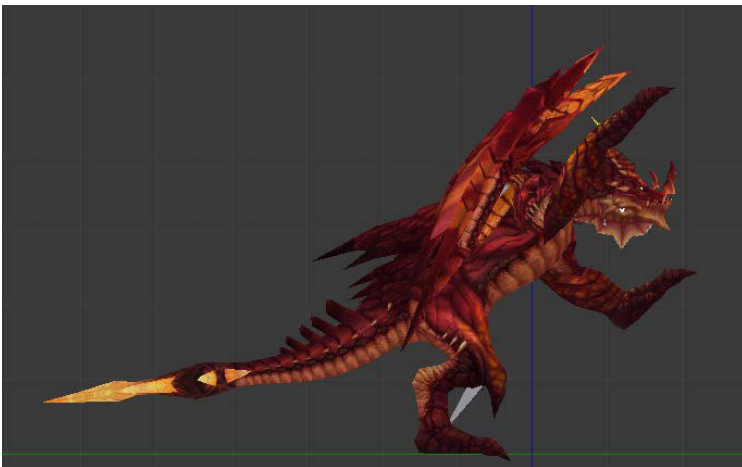
Tampilan animasi serangan normal karakter Dragon dari beberapa sudut pandang ditunjukkan pada Gambar 3.17, dan



Gambar 3.18. Animasi ini dijalankan ketika tombol serangan normal ditekan.



**Gambar 3.17 Tampak kiri animasi serangan normal karakter Dragon**

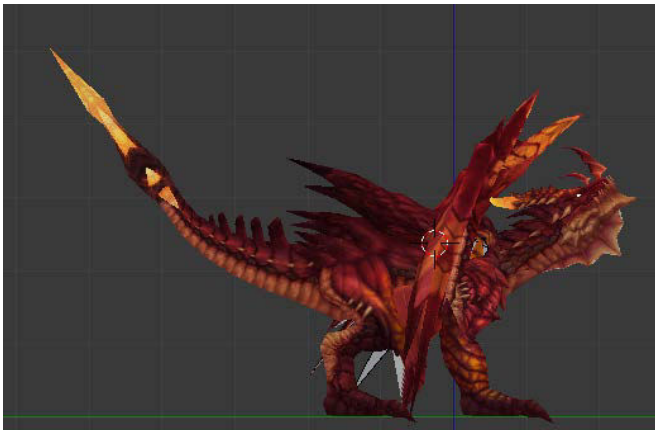


**Gambar 3.18 Tampak kanan animasi serangan normal karakter Dragon**

Tampilan animasi serangan kuat karakter Dragon dari beberapa sudut pandang ditunjukkan pada Gambar 3.19, dan Gambar 3.20. Animasi ini dijalankan ketika tombol serangan kuat ditekan dan nilai MP karakter sama dengan atau lebih dari 30.



**Gambar 3.19 Tampak kiri animasi serangan kuat karakter Dragon**

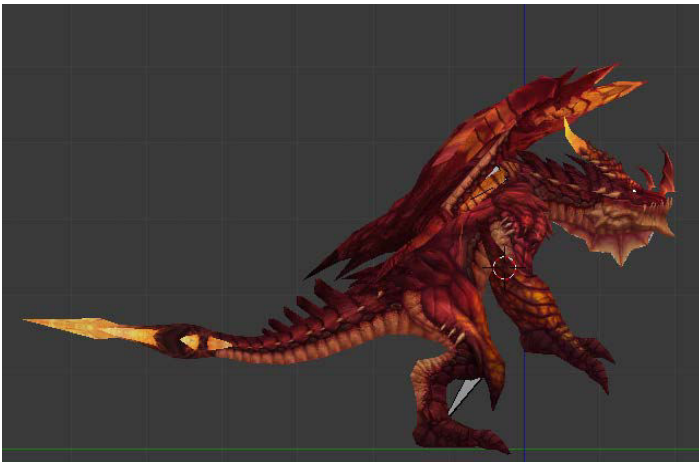


**Gambar 3.20 Tampak kanan animasi serangan kuat karakter Dragon**

Tampilan animasi serangan istimewa karakter Dragon dari beberapa sudut pandang ditunjukkan pada Gambar 3.21, dan Gambar 3.22. Animasi ini dijalankan ketika tombol serangan istimewa ditekan dan nilai MP karakter sama dengan 100.

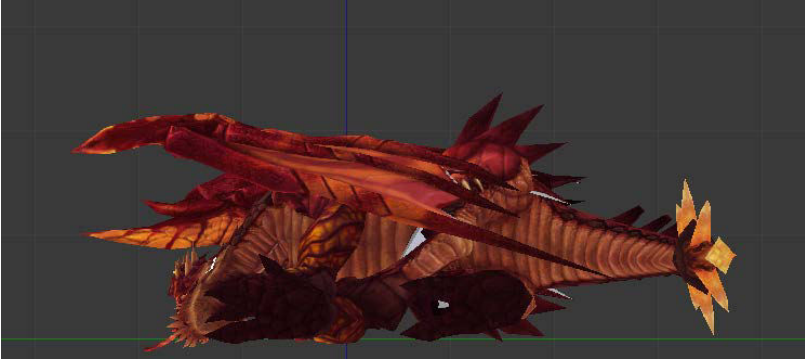


**Gambar 3.21 Tampak kiri animasi serangan istimewa karakter Dragon**



**Gambar 3.22 Tampak kanan animasi serangan istimewa karakter Dragon**

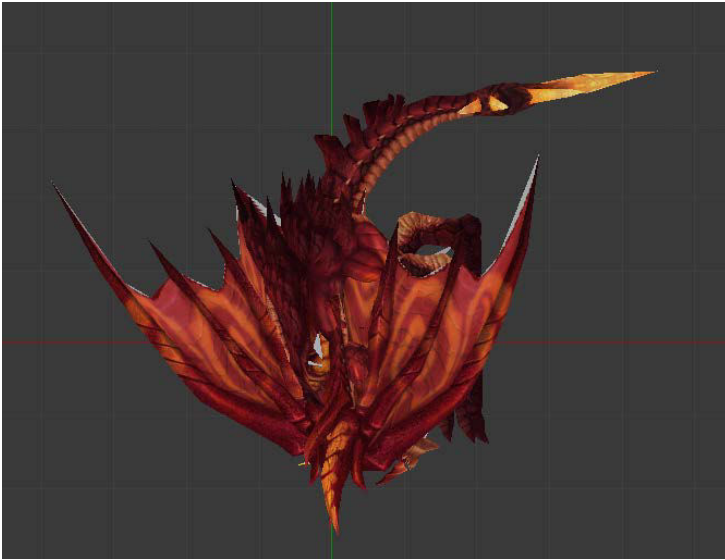
Tampilan animasi jatuh karakter Dragon dari beberapa sudut pandang ditunjukkan pada Gambar 3.23, Gambar 3.24, Gambar 3.25, dan Gambar 3.26. Animasi ini dijalankan jika nilai HP karakter sama dengan 0.



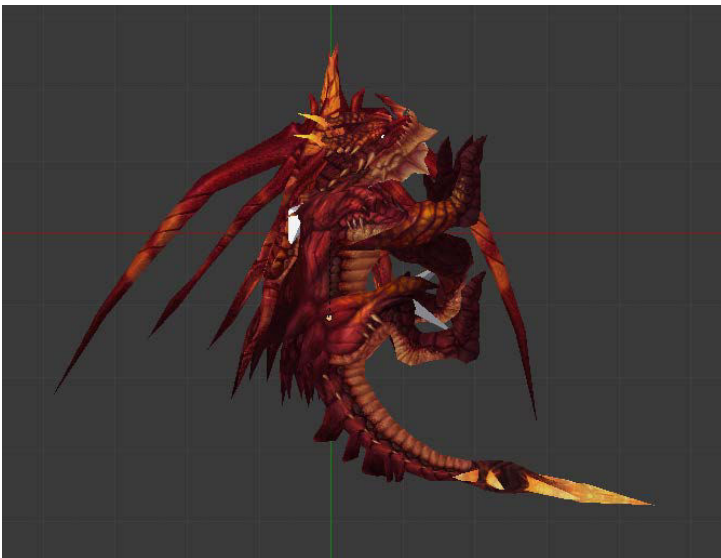
**Gambar 3.23 Tampak kiri animasi jatuh karakter Dragon**



**Gambar 3.24 Tampak kanan animasi jatuh karakter Dragon**



**Gambar 3.25 Tampak atas animasi jatuh karakter Dragon**

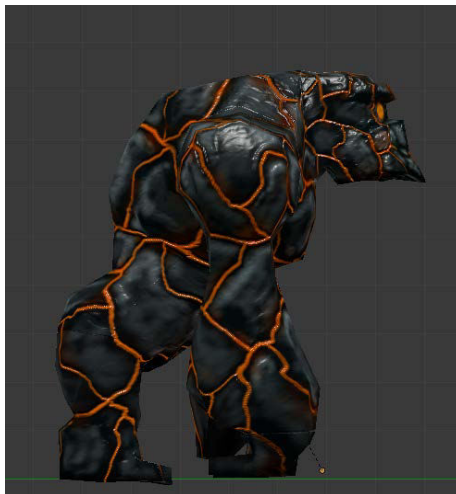


**Gambar 3.26 Tampak bawah animasi jatuh karakter Dragon**

Tampilan karakter Lava Golem dari beberapa sudut pandang ditunjukkan pada Gambar 3.27, Gambar 3.28, Gambar 3.29, Gambar 3.30, Gambar 3.31, dan Gambar 3.32.



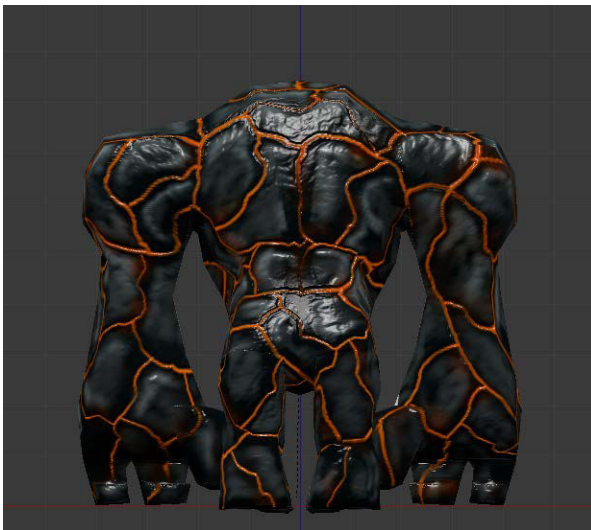
**Gambar 3.27 Tampak kiri karakter Lava Golem**



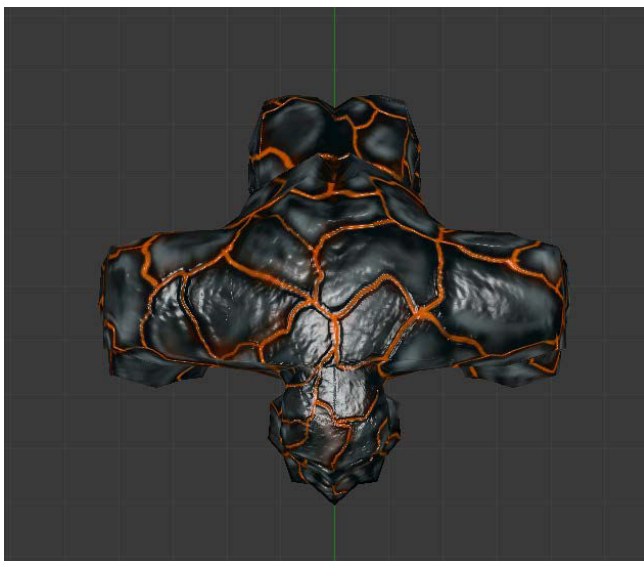
**Gambar 3.28 Tampak kanan karakter Lava Golem**



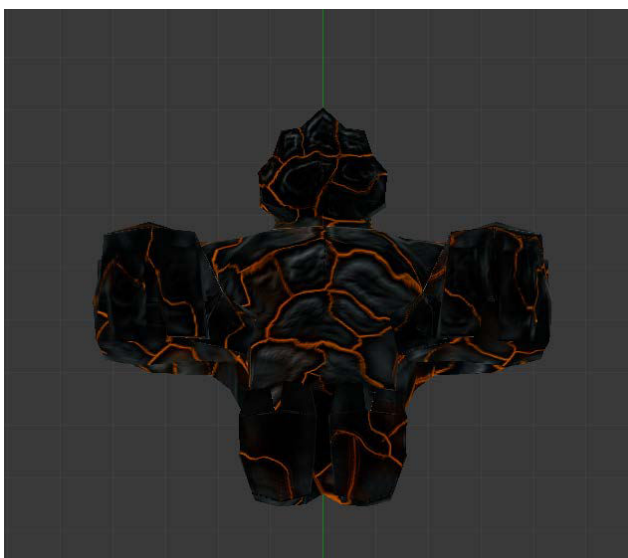
**Gambar 3.29 Tampak depan karakter Lava Golem**



**Gambar 3.30 Tampak belakang karakter Lava Golem**



**Gambar 3.31 Tampak atas karakter Lava Golem**



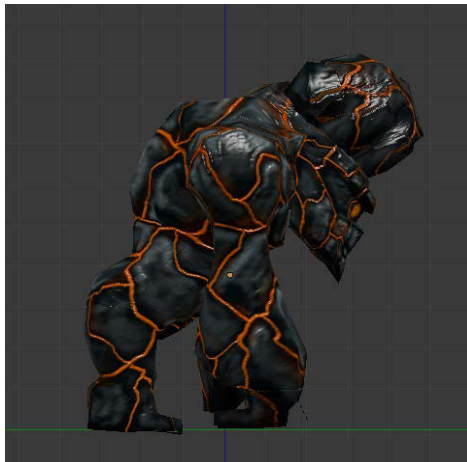
**Gambar 3.32 Tampak bawah karakter Lava Golem**



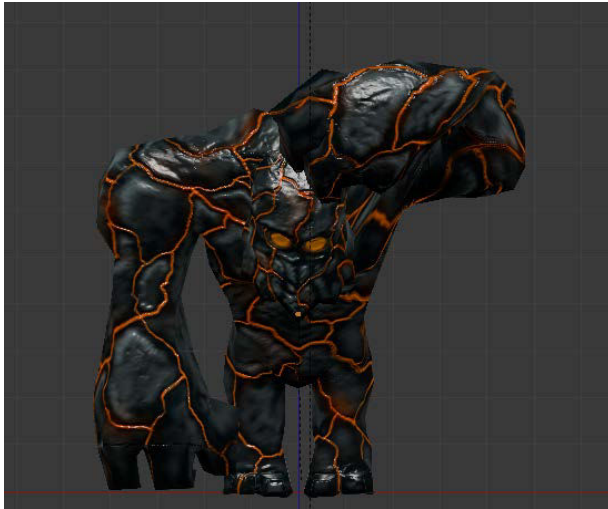
Tampilan animasi siaga (*idle*) karakter Lava Golem dari beberapa sudut pandang ditunjukkan pada Gambar 3.33, Gambar 3.34, Gambar 3.35, dan Gambar 3.36. Animasi ini dijalankan sebagai animasi asal atau *default*.



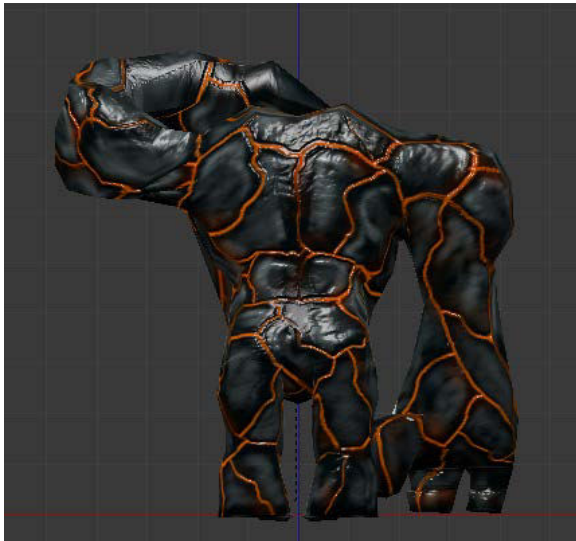
**Gambar 3.33 Tampak kiri animasi siaga karakter Lava Golem**



**Gambar 3.34 Tampak kanan animasi siaga karakter Lava Golem**



**Gambar 3.35 Tampak depan animasi siaga karakter Lava Golem**

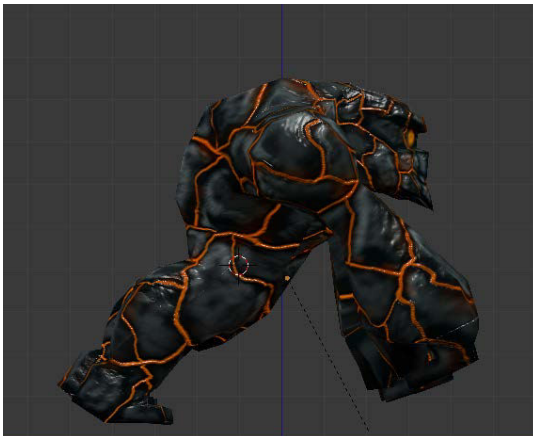


**Gambar 3.36 Tampak belakang animasi siaga karakter Lava Golem**

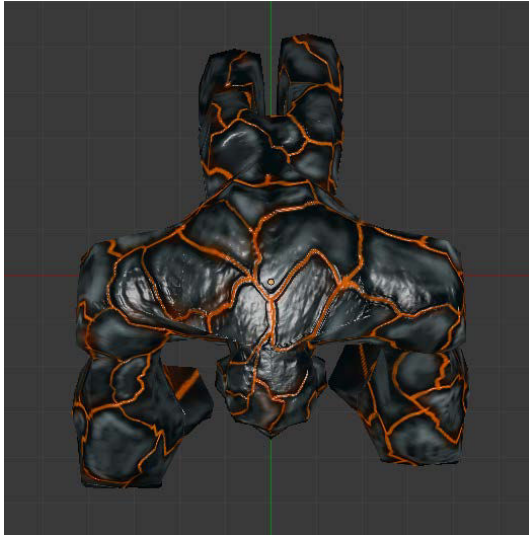
Tampilan animasi berlari karakter Lava Golem dari beberapa sudut pandang ditunjukkan pada Gambar 3.37, Gambar 3.38, Gambar 3.39, dan Gambar 3.40. Animasi ini dijalankan ketika tombol maju atau mundur ditekan dan karakter bertranslasi.



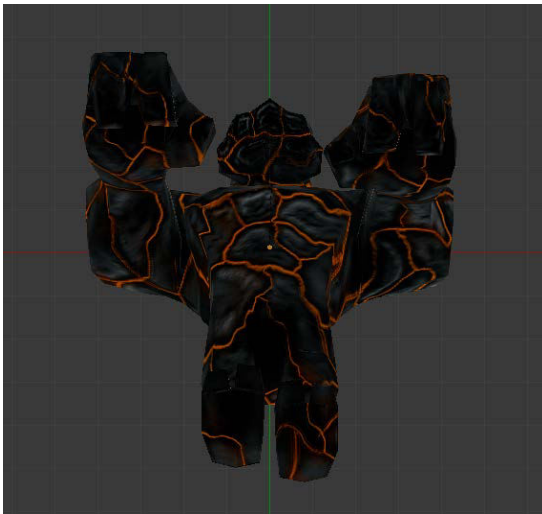
**Gambar 3.37 Tampak kiri animasi berlari karakter Lava Golem**



**Gambar 3.38 Tampak kanan animasi berlari karakter Lava Golem**



**Gambar 3.39 Tampak atas animasi berlari karakter Lava Golem**



**Gambar 3.40 Tampak bawah animasi berlari karakter Lava Golem**

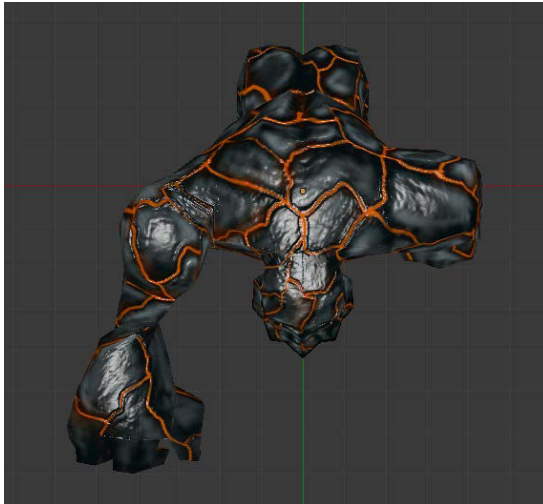
Tampilan animasi serangan normal karakter Lava Golem dari beberapa sudut pandang ditunjukkan pada Gambar 3.41, Gambar 3.42, Gambar 3.43, dan Gambar 3.44. Animasi ini dijalankan ketika tombol serangan normal ditekan.



**Gambar 3.41 Tampak kiri animasi serangan normal karakter Lava Golem**



**Gambar 3.42 Tampak kanan animasi serangan normal karkater Lava Golem**

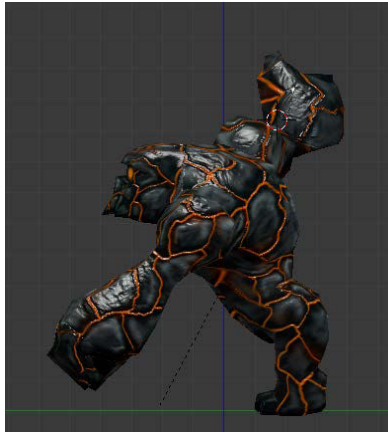


**Gambar 3.43 Tampak atas animasi serangan normal karakter Lava Golem**



**Gambar 3.44 Tampak bawah animasi serangan normal karakter Lava Golem**

Tampilan animasi serangan kuat karakter Lava Golem dari beberapa sudut pandang ditunjukkan pada Gambar 3.45, Gambar 3.46, Gambar 3.47, dan Gambar 3.48. Animasi ini dijalankan ketika tombol serangan kuat ditekan dan nilai MP karakter sama dengan atau lebih dari 30.



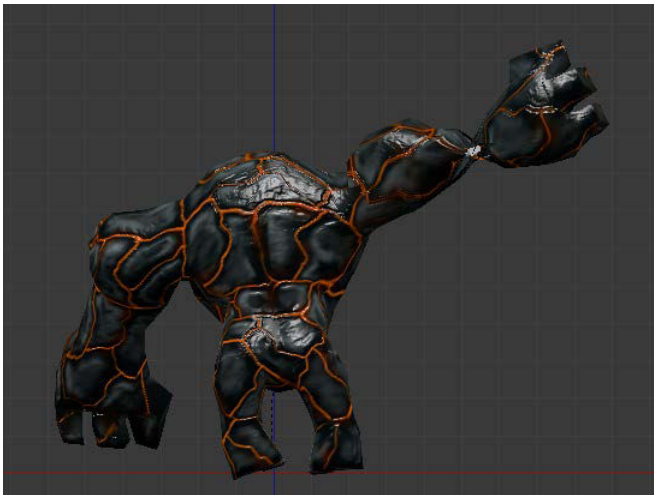
**Gambar 3.45 Tampak kiri animasi serangan kuat karakter Lava Golem**



**Gambar 3.46 Tampak kanan animasi serangan kuat karakter Lava Golem**



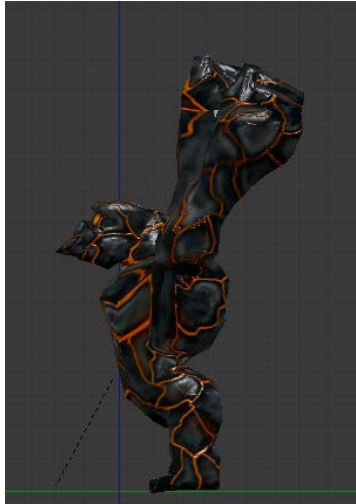
**Gambar 3.47 Tampak depan animasi serangan kuat karakter  
Lava Golem**



**Gambar 3.48 Tampak belakang animasi serangan kuat karakter  
Lava Golem**



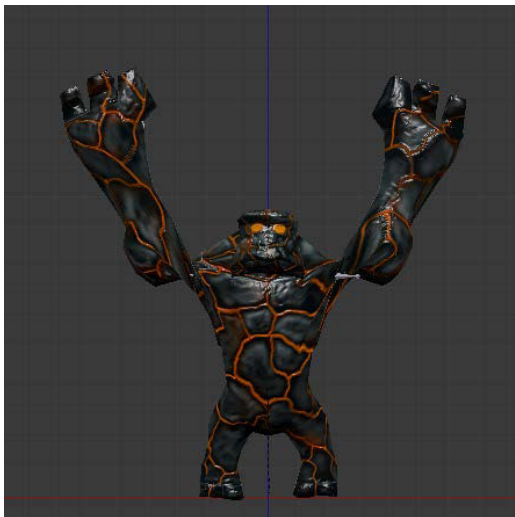
Tampilan animasi serangan istimewa karakter Lava Golem dari beberapa sudut pandang ditunjukkan pada Gambar 3.49, Gambar 3.50, Gambar 3.51, dan Gambar 3.52. Animasi ini dijalankan ketika tombol serangan istimewa ditekan dan nilai MP karakter sama dengan 100.



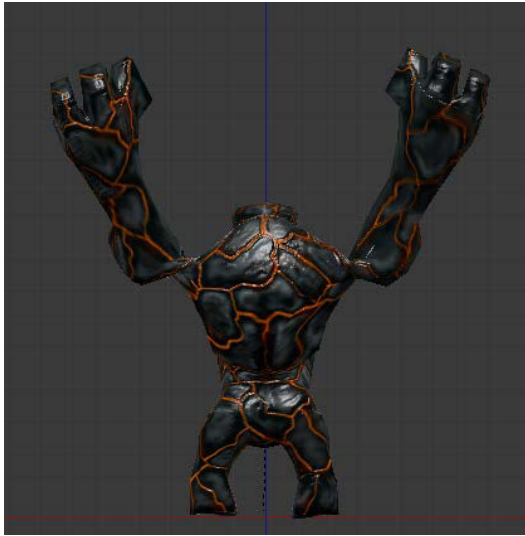
**Gambar 3.49 Tampak kiri animasi serangan istimewa karakter Lava Golem**



**Gambar 3.50 Tampak kanan animasi serangan istimewa karakter Lava Golem**

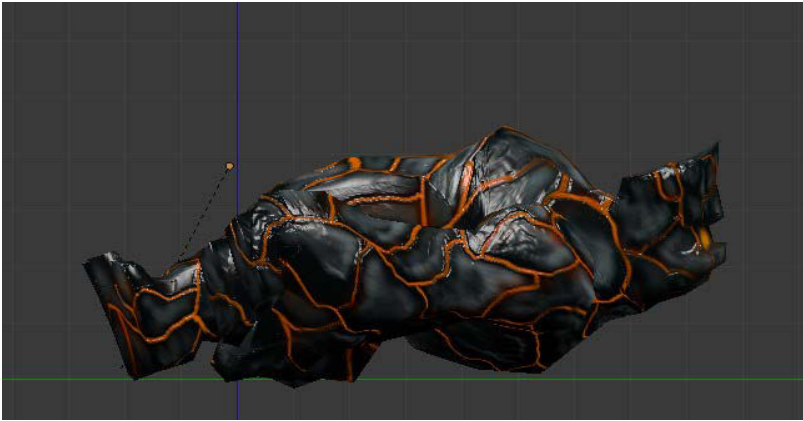


**Gambar 3.51 Tampak depan animasi serangan istimewa karakter Lava Golem**

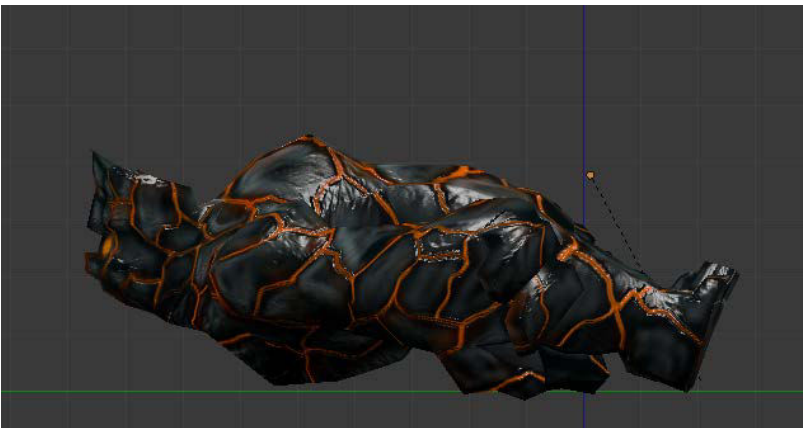


**Gambar 3.52 Tampak belakang animasi serangan istimewa karakter Lava Golem**

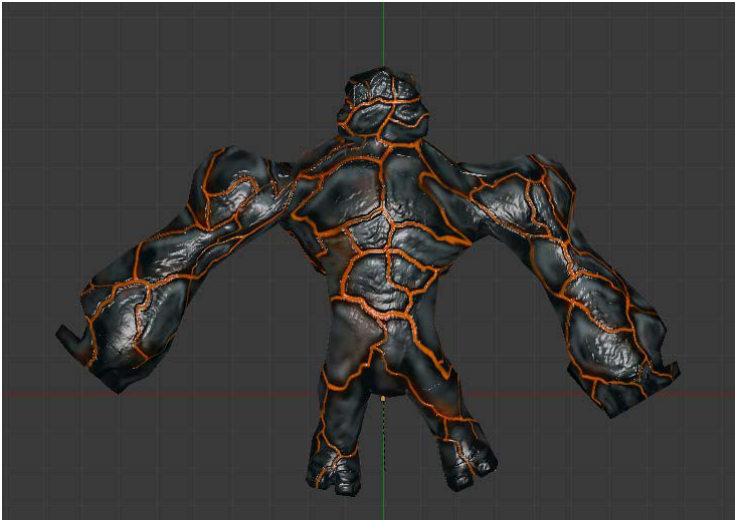
Tampilan animasi jatuh karakter Lava Golem dari beberapa sudut pandang ditunjukkan pada Gambar 3.53, Gambar 3.54, Gambar 3.55, dan Gambar 3.56. Animasi ini dijalankan jika nilai HP karakter sama dengan 0.



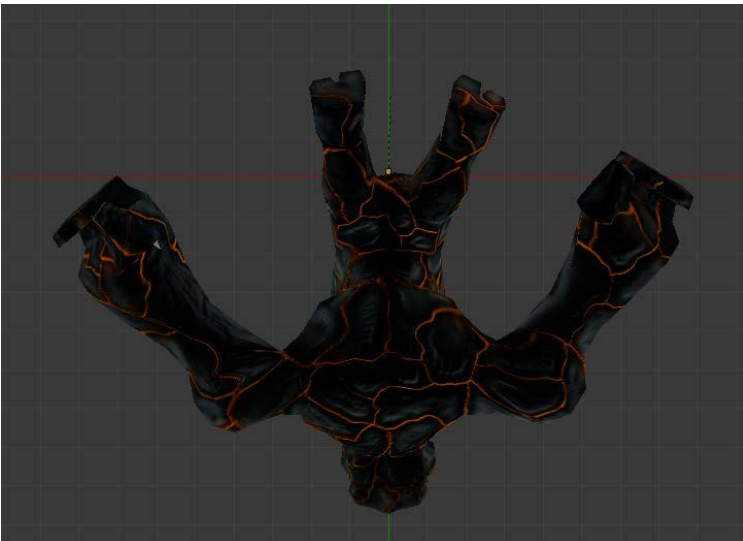
**Gambar 3.53 Tampak kiri animasi jatuh karakter Lava Golem**



**Gambar 3.54 Tampak kiri animasi jatuh karakter Lava Golem**



**Gambar 3.55 Tampak atas animasi jatuh karakter Lava Golem**



**Gambar 3.56 Tampak bawah animasi jatuh karakter Lava Golem**

### 3.3.5 Perancangan Kontrol Permainan

Kontrol dalam permainan ini menggunakan tombol dan *accelerometer* dari perangkat Android yang digunakan.

### 3.3.6 Perancangan Alur Permainan

Permainan ini menceritakan bahwa pemain adalah seseorang yang dapat mengendalikan karakter yang berupa monster atau makhluk. Pemain mengendalikan makhluk untuk bertarung dengan makhluk milik pemain lain dalam duel satu lawan satu. Setiap makhluk memiliki status serangan yang berbeda. Tetapi, setiap makhluk memiliki status HP dan MP yang sama.

Cara memenangkan pertandingan adalah dengan menghabiskan HP karakter lawan terlebih dahulu sebelum HP pemain tersebut dihabiskan oleh lawan. Cara menghabiskan HP lawan adalah dengan menyerang karakter lawan tersebut. Ada 3 jenis serangan, yaitu serangan normal (*Normal Attack*), serangan kuat (*Power Attack*), dan serangan istimewa (*Special Attack*). Pada awal permainan, nilai MP karakter adalah 0. Nilai MP akan bertambah apabila serangan karakter pemain mengenai karakter lawan atau sebaliknya yaitu apabila karakter pemain terkena serangan lawan. *Power Attack* membutuhkan MP sebesar 30 poin dan *Special Attack* membutuhkan 100 poin sehingga pada awal permainan *Power Attack* dan *Special Attack* tidak dapat digunakan.

### 3.3.7 Perancangan Fitur *Augmented Reality*

Fitur *Augmented Reality* (AR) dalam *game* ini menggunakan *package* atau *library* Vuforia. Fitur AR digunakan dalam antarmuka **Battle Menu**. Dalam *game* ini, objek yang akan ditambahkan ke realitas adalah karakter dan arena.

AR pada *package* Vuforia menggantikan kamera pada *game* biasa dengan kamera nyata, misal, *webcam*, kamera ponsel, dll. Apabila kamera yang digunakan menangkap *marker* yang telah ditentukan pada program, realitas akan ditambahi objek karakter dan

arena di atas *marker* tersebut. *Marker* yang digunakan pada *game* ini ditunjukkan pada Gambar 3.57.



**Gambar 3.57 *Marker* untuk fitur AR**

### **3.3.8 Perancangan Fitur *Multiplayer***

Fitur multiplayer dalam *game* ini menggunakan *library* atau *package* Photon Unity Networking. Photon Unity Networking menggunakan sistem *room* supaya karakter para pemain dapat bertemu satu sama lain. Photon Unity Networking menyediakan metode dan *server* untuk menginstansiasi karakter pada jaringan sehingga semua pemain yang tersambung pada satu *room* dapat bertemu.

## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi dari perancangan perangkat lunak. Di dalamnya mencakup proses penerapan dan pengimplementasian algoritma, dan antar muka yang mengacu pada rancangan yang telah dibahas sebelumnya.

### **4.1 Lingkungan Implementasi**

Lingkungan implementasi dari Tugas Akhir ini ditunjukkan pada Tabel 4.1, Tabel 4.2, dan Tabel 4.3.

**Tabel 4.1 Lingkungan Implementasi Perangkat Lunak (1)**

Perangkat Keras	Prosesor: Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz Memori: 2048MB RAM
Perangkat Lunak	Sistem Operasi: Microsoft Windows 8 64-bit Perangkat Pengembang: Unity3D, Blender

**Tabel 4.2 Lingkungan Implementasi Perangkat Lunak (2)**

Perangkat Keras	Prosesor: Intel(R) Core(TM) i3-3240 CPU @ 3.40GHz Memori: 4096MB RAM
Perangkat Lunak	Sistem Operasi: Microsoft Windows 8 64-bit Perangkat Pengembang: Unity3D

**Tabel 4.3 Lingkungan Implementasi Perangkat Lunak (3)**

Perangkat Keras	Prosesor: Intel(R) Atom(TM) Z2520 CPU @ 1.20GHz Memori: 983MB RAM
Perangkat Lunak	Sistem Operasi: Android Versi 4.4.2



## 4.2 Implementasi Alur Proses Aplikasi

Pada subbab ini, akan dibahas implementasi alur proses aplikasi yang telah dibangun pada bab sebelumnya.

### 4.2.1 Implementasi Main Menu

Antarmuka **Main Menu** adalah tampilan pertama setelah aplikasi dibuka. Tampilan ini memiliki dua tombol yaitu “PLAY” dan “EXIT”. Tampilan dari antarmuka ini ditunjukkan pada Gambar 4.1.



**Gambar 4. 1 Implementasi antarmuka Main Menu**

Tombol “PLAY” digunakan untuk masuk ke antarmuka **Character Select Menu**. Tampilan tombol “PLAY” ditunjukkan pada Gambar 4.2.



**Gambar 4.2 Tombol “PLAY”**

Implementasi kode sumber untuk tombol “PLAY” ditunjukkan pada Kode Sumber 4.1.

```
public void ChangeScene (string a)
{
    Application.LoadLevel (a);
}
```

**Kode Sumber 4.1 Tombol “PLAY”**

Tombol “EXIT” digunakan untuk keluar dari aplikasi. Tampilan tombol “EXIT” ditunjukkan pada Gambar 4.3 .



**Gambar 4.3 Tombol “EXIT”**

Implementasi kode sumber untuk tombol “Esc” ditunjukkan pada Kode Sumber 4.2.

```
public void CloseAppButton ()
{
    PlayerPrefs.DeleteKey ("CharacterID");
    Application.Quit ();
}
```

**Kode Sumber 4.2 Tombol “EXIT”**

#### 4.2.2 Implementasi Character Select Menu

Antarmuka **Character Select Menu** adalah antarmuka yang tampil setelah tombol “PLAY” pada antarmuka **Main Menu** ditekan. Antarmuka ini digunakan untuk memilih karakter yang hendak digunakan untuk bermain. Untuk memilih karakter, pemain harus menekan tombol yang sesuai dengan karakter yang bersangkutan. Tampilan karakter akan ditayangkan sesuai dengan tombol yang dipilih. Dalam *game* ini terdapat 2 pilihan karakter yaitu Dragon dan Lava Golem.

Karakter *default* yang terpilih adalah Dragon. Sehingga, tampilan karakter yang ditayangkan pertama kali saat memasuki **Character Select Menu** adalah Dragon. Tampilan tombol karakter Dragon ditunjukkan pada Gambar 4.4 sedangkan tombol karakter Lava Golem ditunjukkan pada Gambar 4.5. Karakter yang sedang ditampilkan akan berputar lambat searah jarum jam. Pengguna dapat menggeser (*slide*) layar ke arah kanan atau kiri untuk melihat tampilan karakter dari sudut lain.



**Gambar 4.4 Implementasi antarmuka Character Select Menu – Pilihan karakter Dragon**

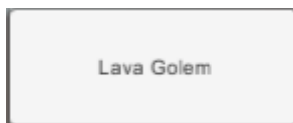


**Gambar 4.5 Implementasi antarmuka Character Select Menu – Pilihan karakter Lava Golem**

Tampilan dari tombol setiap karakter ditunjukkan pada Gambar 4.6 dan Gambar 4.7.



**Gambar 4.6 Tombol karakter Dragon**



**Gambar 4.7 Tombol Karakter Lava Golem**

Kode sumber untuk memilih karakter ditunjukkan pada Kode Sumber 4.3.

```

private List<GameObject> characters;
public int characterIndex = 0;

CharacterInstantiation characterInstantiation;

void Start ()
{
    characters = new List<GameObject> ();

    foreach (Transform t in transform)
    {
        characters.Add(t.gameObject);
        t.gameObject.SetActive(false);
    }
    characters [characterIndex].SetActive (true);
}

private void Update ()
{
    transform.Rotate(0, 20 * Time.deltaTime, 0);
    if(Input.GetMouseButton(0))
        transform.Rotate (new Vector3(0.0f, (Input.GetAxis ("Mouse X")) * -3));
}

```

### Kode Sumber 4.3 Pemilihan karakter

Kode sumber untuk tiap tombol karakter ditunjukkan pada Kode Sumber 4.4.

```

public void Select(int index)
{
    if (index == characterIndex)
        return;
    if (index < 0 || index >= characters.Count)
        return;

    characters [characterIndex].SetActive (false);
    characterIndex = index;
    characters [characterIndex].SetActive (true);
}

```

### Kode Sumber 4.4 Tombol karakter

Setelah kedua pemain selesai memilih karakter yang dikehendaki, kedua pemain menekan tombol “FIGHT!” untuk masuk ke antarmuka **Battle Menu** untuk bermain dengan karakter yang telah dipilih. Karakter yang telah dipilih oleh kedua pemain akan diinstansiasi pada arena yang telah tersedia dan kedua pemain akan disambungkan dengan *server* Photon. Tampilan tombol “FIGHT!” ditunjukkan pada Gambar 4.8.



**Gambar 4.8 Tombol “FIGHT!”**

Kode sumber untuk tombol “FIGHT!” ditunjukkan pada Kode Sumber 4.5.

```
public void LetsBattle(string a)
{
    string VERSION = "v0.0.1";
    Application.LoadLevel (a);
    PhotonNetwork.ConnectUsingSettings (VERSION);
}
```

**Kode Sumber 4.5 Tombol “FIGHT!”**

Pemain dapat kembali ke antarmuka **Main Menu** dengan menekan tombol “Esc”. Tampilan tombol “Esc” ditunjukkan pada Gambar 4.6.



**Gambar 4.9 Tombol “Esc” pada antarmuka Character Select Menu**

Kode sumber untuk tombol “Esc” ditunjukkan pada Kode Sumber 4.6.

```
public void ChangeScene (string a)
{
    Application.LoadLevel (a);
}
```

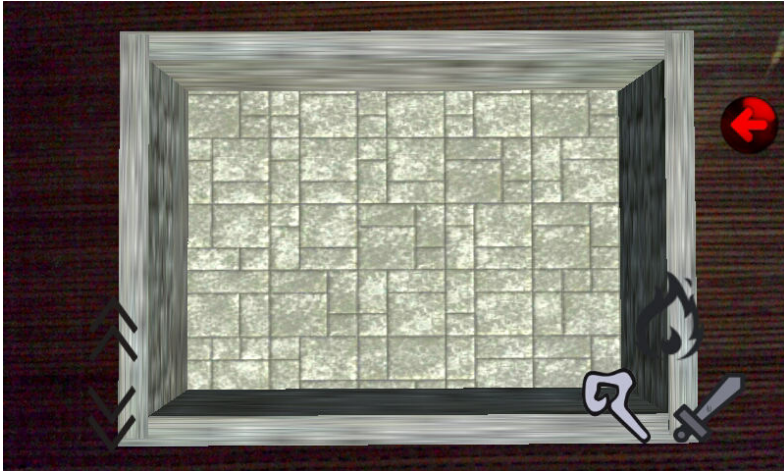
**Kode Sumber 4.6 Tombol “Esc”**

#### **4.2.3 Implementasi Battle Menu**

Antarmuka **Battle Menu** adalah antarmuka yang tampil setelah tombol “FIGHT!” (tombol dengan gambar dua pedang yang saling bersilang) pada antarmuka **Character Select Menu** ditekan. Antarmuka ini digunakan untuk melakukan kasus penggunaan UC-002 Bermain.

Implementasi antarmuka **Battle Menu** sedikit berbeda dengan perancangannya. Dalam implementasi antarmuka ini, ditambahkan dua tombol arah yaitu arah maju dan mundur. Kedua tombol ini digunakan untuk menggerakkan karakter ke depan atau belakang berdasarkan kemana arah karakter menghadap. Implementasi antarmuka **Battle Menu** ditunjukkan pada Gambar 4.10, Gambar 4.11, dan Gambar 4.12. Gambar-gambar tersebut menunjukkan bahwa

*marker* telah terbaca sehingga objek arena dan karakter tampil pada layar.



**Gambar 4. 10 Implementasi antarmuka Battle Menu – karakter belum terinstansiasi**



**Gambar 4.11 Implementasi antarmuka Battle Menu – satu karakter telah terinstansiasi**





**Gambar 4.12 Implementasi antarmuka Battle Menu – dua karakter telah terinstansiasi**

#### **4.2.4 Implementasi Kontrol Permainan**

Perintah kepada karakter dibagi menjadi dua jenis, yaitu gerakan dan serangan. Perintah gerakan juga dibagi menjadi dua jenis yaitu menggunakan tombol dan *accelerometer*. Ketiga perintah serangan diakomodasi oleh tombol. Semua tombol ini baru akan berfungsi apabila karakter yang dipilih pemain telah terinstansiasi pada antarmuka **Battle Menu**.

Perintah gerakan dibagi menjadi dua jenis, yaitu menggunakan tombol dan *accelerometer*. Untuk menggerakkan karakter ke depan atau belakang, pemain menggunakan tombol dan untuk memutar karakter, pemain menggunakan *accelerometer*. Penjelasan lebih rinci tentang implementasi perintah gerakan karakter adalah sebagai berikut:

1. Tombol maju

Tombol maju digunakan untuk menggerakkan karakter ke arah depan. Selama tombol ini ditekan,

karakter akan terus bergerak maju dan menjalankan animasi berlari. Tampilan tombol maju ditunjukkan pada Gambar 4.13 dan kode sumber untuk tombol ini ditunjukkan pada Kode Sumber 4.7.



**Gambar 4.13 Tombol maju**

```
public void PositionUpButton ()
{
    NM.SelectedCharacter.transform.Translate (Vector3.forward * Time.deltaTime * TranslationSpeed);
    Animator temp = NM.SelectedCharacter.GetComponent<Animator> ();
    temp.SetBool ("Run", true);
}
```

**Kode Sumber 4.7 Tombol maju**

## 2. Tombol mundur

Tombol mundur digunakan untuk menggerakkan karakter ke arah belakang. Selama tombol ini ditekan, karakter akan terus bergerak ke arah belakang dan menjalankan animasi berlari. Dalam permainan, tombol ini dapat digunakan untuk menghindari serangan lawan. Tampilan tombol mundur ditunjukkan pada Gambar 4.14

dan kode sumber untuk tombol ini ditunjukkan pada Kode Sumber 4.8.



**Gambar 4.14 Tombol mundur**

```
public void PositionDownButton ()
{
    NM.SelectedCharacter.transform.Translate (-Vector3.forward * Time.deltaTime * TranslationSpeed);
    Animator temp = NM.SelectedCharacter.GetComponent<Animator> ();
    temp.SetBool ("Run", true);
}
```

#### **Kode Sumber 4.8 Tombol mundur**

3. Berputar dengan masukan *accelerometer*  
 Selain gerakan maju dan mundur, karakter juga dapat berotasi untuk mengubah arah. Untuk merotasi karakter ke arah kanan, pemain harus memiringkan perangkatnya ke arah kanan. Untuk merotasi karakter ke arah kiri, pemain harus memiringkan perangkatnya ke arah kiri. Kode sumber untuk pengontrolan karakter dengan

*accelerometer* ditunjukkan pada Kode Sumber 4.9.

```
void Update ()
{
    transform.Translate (Input.acceleration.x, 0, 0);
    transform.Rotate (0, Input.acceleration.x * 20.0f, 0, Space.World);
}
```

#### **Kode Sumber 4.9 Perintah rotasi dengan *accelerometer***

Perintah serangan dibagi menjadi 3 jenis, yaitu serangan normal (*Normal Attack*), serangan kuat (*Power Attack*), dan serangan istimewa (*Special Attack*). Jika serangan mengenai karakter lawan, karakter lawan akan terkena *damage* yang membuat *Health Point* (HP) karakter lawan berkurang. Jumlah *damage* bergantung pada status masing-masing karakter yang digunakan oleh pemain seperti yang telah dijelaskan pada bab sebelumnya. Penjelasan lebih rinci tentang tombol-tombol perintah serangan adalah sebagai berikut:

1. Tombol “Attack”

Tombol “Attack” digunakan untuk melakukan serangan normal kepada karakter lawan. Serangan ini tidak membutuhkan MP, jangkauannya relatif dekat, dan *damage* yang dihasilkan relatif kecil. Tampilan tombol “Attack” ditunjukkan pada Gambar 4.15 dan kode sumber untuk tombol ini ditunjukkan pada Kode Sumber 4.10.



**Gambar 4.15 Tombol “Attack”**

```
public void NormalAttack()
{
    Ray ray = new Ray (GameObject.Find ("AttackingPoint").transform.position, GameObject.Find ("AttackingPoint").transform.forward);
    RaycastHit hitInfo;

    if (Physics.Raycast (ray, out hitInfo, 40))
    {
        if(hitInfo.collider.tag == "Character")
        {
            CharacterHealth temp = hitInfo.transform.GetComponent<CharacterHealth>();
            temp.GetComponent<PhotonView>().RPC("TakeDamage", PhotonTargets.All, NormalAttackDamage);
            temp.GetComponent<PhotonView>().RPC("TakeMP", PhotonTargets.All, 2);

            CH.GetComponent<PhotonView>().RPC("TakeMP", PhotonTargets.All, 3);

            HitSound.Play();
        }
    }
}
```

**Kode Sumber 4.10 Tombol “Attack”**

## 2. Tombol “Power Attack”

Tombol “Power Attack” digunakan untuk melakukan serangan kuat kepada karakter lawan. Serangan ini membutuhkan *Mental Point* (MP) sebanyak 30 poin, jangkauannya relatif jauh, dan *damage* yang dihasilkan cukup besar. Tampilan tombol “Power Attack” ditunjukkan pada Gambar 4.16 dan kode sumber untuk tombol ini ditunjukkan pada Kode Sumber 4.11.



**Gambar 4.16 Tombol “Power Attack”**

```
public void PowerAttack()
{
    Ray ray = new Ray (GameObject.Find ("AttackingPoint").transform.position, GameObject.Find ("AttackingPoint").transform.forward);
    RaycastHit hitInfo;

    if (Physics.Raycast (ray, out hitInfo, 400))
    {
        if (hitInfo.collider.tag == "Character")
        {
            CharacterHealth temp = hitInfo.transform.GetComponent<CharacterHealth>();
            temp.GetComponent<PhotonView>().RPC("TakeDamage", PhotonTargets.All, PowerAttackDamage);
            temp.GetComponent<PhotonView>().RPC("TakeMP", PhotonTargets.All, 12);

            temp.GetComponent<PhotonView>().RPC("Attacked", PhotonTargets.All);

            Explosion.Play();
        }
    }

    CH.CurMP -= 30;
}
```

**Kode Sumber 4.11 Tombol “Power Attack”**

3. Tombol “Special Attack”  
Tombol “Special Attack” digunakan untuk melakukan serangan istimewa kepada karakter lawan. Serangan ini membutuhkan *Mental Point* (MP) sebanyak 100 poin (*full bar*), jangkauannya paling jauh dari semua serangan, dan *damage* yang dihasilkan paling besar dari

semua serangan. Tampilan tombol “Special Attack” ditunjukkan pada Gambar 4.17 dan kode sumber untuk tombol ini ditunjukkan pada Kode Sumber 4.12.



**Gambar 4.17 Tombol “Special Attack”**

```
public void SpecialAttack()
{
    Ray ray = new Ray (GameObject.Find ("AttackingPoint").transform.position, GameObject.Find ("AttackingPoint").transform.forward);
    RaycastHit hitInfo;

    if (Physics.Raycast (ray, out hitInfo, 500))
    {
        if (hitInfo.collider.tag == "Character")
        {
            CharacterHealth temp = hitInfo.transform.GetComponent<CharacterHealth>();
            temp.GetComponent<PhotonView>().RPC("TakeDamage", PhotonTargets.All, SpecialAttackDamage);
            temp.GetComponent<PhotonView>().RPC("TakeMP", PhotonTargets.All, 18);

            temp.GetComponent<PhotonView>().RPC("Attacked", PhotonTargets.All);

            Explosion.Play();
        }
    }

    CH.CurMP -= 100;
}
```

**Kode Sumber 4.12 Tombol “Special Attack”**

Antarmuka **Battle Menu** memiliki tombol “Esc” untuk kembali ke **Character Select Menu**. Tombol “Esc” pada antarmuka ini berbeda dengan tombol “Esc” pada **Character**

**Select Menu.** Karakter yang telah terinstansiasi akan dihilangkan dari antarmuka **Battle Menu** dan sambungan pemain dengan *server* Photon akan diputus.

Tombol “Esc” digunakan untuk kembali ke antarmuka **Character Select Menu**. Karakter yang telah terinstansiasi akan dihilangkan dari antarmuka **Battle Menu** dan sambungan pemain dengan *server* Photon akan diputus. Tampilan tombol “Esc” ditunjukkan pada Gambar 4.18 dan kode sumber untuk tombol ini ditunjukkan pada Kode Sumber 4.13.



**Gambar 4.18 Tombol “Esc” pada antarmuka Battle Menu**

```
public void EscapeToCharacterSelect ()
{
    Application.LoadLevel ("Scene_CharacterSelect");
    PhotonNetwork.Disconnect ();
}
```

**Kode Sumber 4.13 Tombol “Esc” pada antarmuka Character Select Menu**

#### 4.2.5 Implementasi *Multiplayer*

Untuk mengakomodasi fitur *multiplayer*, *game* ini menggunakan layanan *server* Photon secara *online*. Sehingga, kedua pemain dapat terhubung dan terinstansiasi pada *room* yang sama. Karena terhubung secara *online*, kedua pemain tidak harus benar-benar di dalam lokasi yang sama. Tetapi, tiap-tiap pemain



harus memiliki *marker* masing-masing. Implementasi pembuatan *room* untuk *game* ini pada *server* Photon ditunjukkan pada Kode Sumber 4.14.

```
void OnJoinedLobby ()
{
    RoomOptions roomOptions = new RoomOptions() { isVisible = false, maxPlayers = 2 };
    PhotonNetwork.JoinOrCreateRoom(RoomName, roomOptions, TypedLobby.Default);
}
```

#### Kode Sumber 4.14 Pembuatan *room* pada *server* Photon

Implementasi instansiasi karakter pada arena ditunjukkan pada Kode Sumber 4.15.

```
void OnJoinedRoom ()
{
    if (PhotonNetwork.countOfPlayersInRooms <= 0)
    {
        SelectedCharacter = (GameObject) PhotonNetwork.Instantiate
            (CharacterName [SelectedCharacterIndex],
             P1SpawnPoint.localPosition,
             P1SpawnPoint.localRotation, 0);
    }
    else if (PhotonNetwork.countOfPlayersInRooms >= 1)
    {
        SelectedCharacter = (GameObject) PhotonNetwork.Instantiate
            (CharacterName [SelectedCharacterIndex],
             P2SpawnPoint.localPosition,
             P2SpawnPoint.localRotation, 0);
    }

    SelectedCharacter.transform.SetParent(ImageTarget.transform);
    SelectedCharacter.GetComponent<MeleeSystem> ().enabled = true;
    SelectedCharacter.GetComponent<CharacterHealth> ().enabled = true;
    SelectedCharacter.GetComponent<AccelerometerMovement> ().enabled = true;
}
```

#### Kode Sumber 4.15 Instansiasi karakter pada arena

### 4.2.6 Implementasi Pertandingan

Kedua pemain yang karakternya telah terinstansiasi pada arena pada antarmuka **Battle Menu** dapat memulai permainan.

Pemain dapat menggerakkan karakternya ke depan dengan tombol maju atau ke belakang dengan tombol mundur. Pemain juga dapat memutar posisi karakternya ke arah kanan dengan cara memiringkan perangkatnya ke kanan atau memutar posisi karakternya ke arah kiri dengan cara memiringkan perangkatnya ke kiri.

Setiap serangan memiliki jarak tertentu sehingga mengenai karakter lawan. Pemain harus mengatur posisi karakternya agar serangannya berhasil.

Pada awal permainan, serangan yang dapat digunakan hanya serangan normal. Setelah MP karakter terkumpul, pemain dapat menggunakannya untuk melakukan serangan kuat atau serangan istimewa.

Setelah karakter terinstansiasi, pemain dapat melihat *bar* status HP dan MP di sebelah kiri atas layar. Kode Sumber yang berisi status HP dan MP *default* dari tiap karakter ditunjukkan pada Kode Sumber 4.16. Jumlah HP maksimal adalah 1000 dan jumlah MP maksimal adalah 100.

```
public int MaxHealth = 1000;  
public int CurHealth = 1000;  
public int MaxMP = 100;  
public int CurMP = 0;
```

#### **Kode Sumber 4.16 Status HP dan MP *default* karakter**

Kode sumber yang digunakan untuk mengatur pengurangan HP dan penambahan MP ditunjukkan pada Kode Sumber 4.17.

```

[PunRPC]
public void TakeDamage(int adj)
{
    CurHealth -= adj;

    if (CurHealth < 0)
    {
        CurHealth = 0;
    }

    if (CurHealth <= 0)
    {
        Lose();
    }

    HealthBarLength = (Screen.width / 3) * (CurHealth / (float)MaxHealth);
}

[PunRPC]
public void TakeMP(int adj)
{
    CurMP += adj;

    if (CurMP > 100)
        CurMP = 100;

    MPBarLength = (Screen.width / 3) * (CurMP / (float)MaxMP);
}

```

#### **Kode Sumber 4.17 Pengaturan perubahan status HP dan MP**

Kode sumber untuk menayangkan status HP dan MP karakter ditunjukkan pada Kode Sumber 4.18.

```

void OnGUI ()
{
    GUI.Box (new Rect (30, 10, HealthBarLength, 20), "HP");
    GUI.Box (new Rect (30, 35, MPBarLength, 20), "MP");
}

```

#### **Kode Sumber 4.18 Menampilkan status HP dan MP**

Apabila terdapat salah satu karakter yang nilai HP-nya nol atau habis karena terkena serangan dari lawan, maka pemain

yang HP karakternya habis tersebut dinyatakan kalah dan pemain yang berhasil menghabiskan HP karakter lawan dinyatakan sebagai pemenang. Karakter yang HP-nya habis tersebut akan menjalankan animasi jatuh. Kemudian, sistem akan menampilkan notifikasi atau *popup* “GAME OVER” yang menyatakan bahwa permainan telah selesai. Setelah itu, antarmuka “Character Select” akan tampil dan sambungan kedua pemain dengan *server* Photon akan diputus. Setelah masuk ke antarmuka **Character Select**, kedua pemain bisa memilih karakter yang dikehendaki dan bermain lagi. Implementasi ini ditunjukkan pada Kode Sumber 4.19.

```
void LookForGameObject ()
{
    foreach (GameObject go in GameObject.FindGameObjectsWithTag("Character"))
    {
        if(go.GetComponent<CharacterHealth>().CurHealth <= 0)
        {
            Theme.Stop();

            Delay1 -= Time.deltaTime;

            if(Delay1 <= 0)
            {
                PopUp.enabled = true;
                Delay2 -= Time.deltaTime;
                if(Delay2 <= 0)
                {
                    Application.LoadLevel("Scene_CharacterSelect");
                    PhotonNetwork.Disconnect ();
                }
            }
        }
    }
}
```

**Kode Sumber 4.19 Notifikasi (*popup*) dan pengembalian pemain ke antarmuka Character Select**

## BAB V

### PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan. Proses pengujian dilakukan menggunakan metode *black-box* berdasarkan skenario yang telah ditentukan dan pengujian dilakukan dengan survei langsung kepada pengguna.

#### 5.1 Lingkungan Uji Coba

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini ditunjukkan pada Tabel 5.1, Tabel 5.2, dan Tabel 5.3.

**Tabel 5.1 Lingkungan Uji Coba Perangkat Lunak (1)**

Perangkat Keras	Prosesor: Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz Memori: 2048MB RAM
Perangkat Lunak	Sistem Operasi: Microsoft Windows 8 64-bit Perangkat Pengembang: Unity3D, Blender

**Tabel 5.2 Lingkungan Uji Coba Perangkat Lunak (2)**

Perangkat Keras	Prosesor: Intel(R) Core(TM) i3-3240 CPU @ 3.40GHz Memori: 4096MB RAM
Perangkat Lunak	Sistem Operasi: Microsoft Windows 8 64-bit Perangkat Pengembang: Unity3D

**Tabel 5.3 Lingkungan Uji Coba Perangkat Lunak (3)**

Perangkat Keras	Prosesor: Intel(R) Atom(TM) Z2520 CPU @ 1.20GHz Memori: 983MB RAM
Perangkat Lunak	Sistem Operasi: Android Versi 4.4.2

## 5.2 Skenario Pengujian Fungsionalitas

Pengujian ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi pada tahap kebutuhan benar-benar diimplementasikan dan bekerja semestinya. Selain itu, langkah ini ditujukan untuk mengetahui kesesuaian keluaran dari setiap tahapan atau langkah penggunaan fitur terhadap skenario yang dipersiapkan. Pengujian dilakukan dengan menggunakan metode *black-box*.

### 5.2.1 Skenario Pengujian Pemilihan Karakter

Skenario pengujian pemilihan karakter digunakan untuk mengetahui apakah fungsionalitas pemilihan karakter sudah berjalan sesuai dengan apa yang diharapkan. Skenario ini ditunjukkan pada Tabel 5.4.

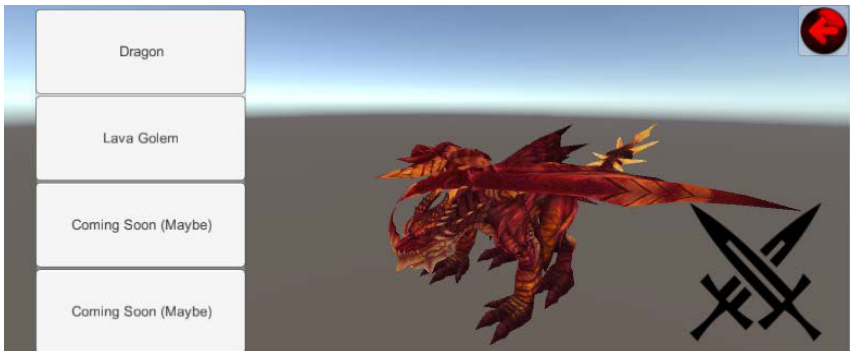
**Tabel 5.4 Pengujian Pemilihan Karakter**

	UC-001
Kondisi Awal	Pengguna memasuki antarmuka <b>Character Select Menu</b>
Prosedur Pengujian	Pengguna memilih karakter dengan cara menekan tombol karakter dan menggeser ( <i>slide</i> ) layar ke arah kanan atau kiri untuk melihat tampilan karakter dari sudut lain

Hasil yang diharapkan	Pengguna dapat melihat dan memilih karakter sesuai dengan yang dikehendaki
Hasil yang diperoleh	Pengguna dapat melihat dan memilih karakter sesuai dengan yang dikehendaki
Kesimpulan.	Pengujian berhasil

#### 5.2.1.1 PF-01: Pengujian Antarmuka Character Select Menu

Pengujian dimulai ketika pemain memasuki antarmuka **Character Select Menu** seperti yang ditunjukkan pada Gambar 5.1. Pemain memilih karakter dengan menekan tombol karakter yang tersedia. Pemain juga dapat menggeser (*slide*) layar ke arah kanan atau kiri untuk memutar karakter supaya pemain dapat melihat tampilan karakter dari sudut lain. Terdapat dua karakter dalam *game* ini, yaitu Dragon dan Lava Golem.



**Gambar 5.1 Antarmuka Character Select Menu – pilihan karakter Dragon**



**Gambar 5.2 Antarmuka Character Select Menu – pilihan karakter Lava Golem**

### 5.2.2 Skenario Pengujian Permainan

Skenario pengujian permainan digunakan untuk mengetahui apakah fungsionalitas bermain sudah berjalan sesuai dengan apa yang diharapkan. Skenario ini ditunjukkan pada Tabel 5.5.

**Tabel 5.5 Pengujian Permainan**

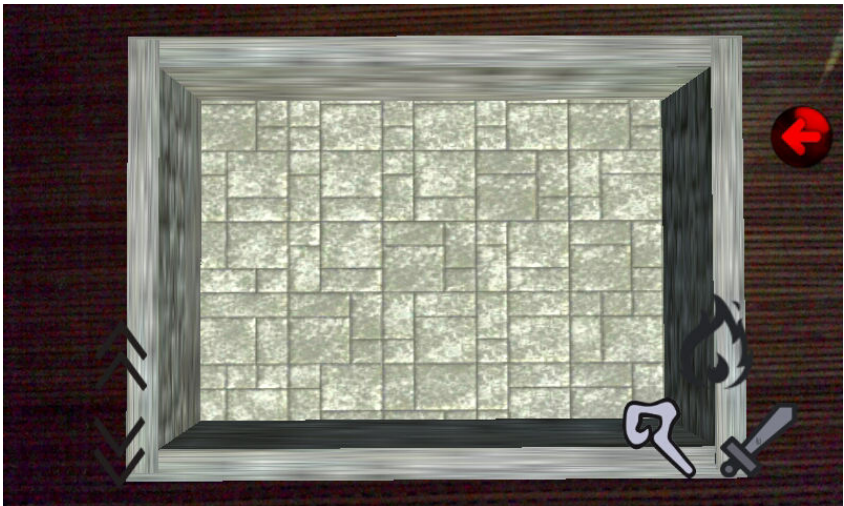
	<b>UC-002</b>
Kondisi Awal	Pengguna berada pada antarmuka <b>Character Select Menu</b>
Prosedur Pengujian	Pengguna menekan tombol “FIGHT!” supaya karakter yang telah dipilih terinstansiasi pada arena pada antarmuka <b>Battle Menu</b>
Hasil yang diharapkan	Pengguna merasa terhibur dengan fitur-fitur yang ditawarkan dan semua fungsionalitas berjalan dengan lancar



Hasil yang diperoleh	Pengguna merasa terhibur dengan fitur-fitur yang ditawarkan dan semua fungsionalitas berjalan lancar
Kesimpulan.	Pengujian berhasil

#### 5.2.2.1 PF-02: Pengujian Instansiasi Karakter pada Battle Menu

Pengujian dimulai ketika pemain memasuki antarmuka **Battle Menu** seperti yang ditunjukkan pada Gambar 5.3.



**Gambar 5.3 Antarmuka Battle Menu – karakter belum terinstansiasi**

Gambar 5.3 menunjukkan bahwa karakter yang dipilih oleh pemain belum terinstansiasi dan menunggu sambungan ke *server* Photon. Setelah sambungan ke *server* Photon berhasil dilakukan, karakter akan terinstansiasi ke arena seperti yang ditunjukkan oleh Gambar 5.4 dan Gambar 5.5.



**Gambar 5.4 Antarmuka Battle Menu – satu karakter telah terinstansiasi**

Gambar 5.4 menunjukkan bahwa hanya satu karakter yang telah terinstansiasi. Hal ini bisa terjadi karena beberapa sebab, diantaranya karena pemain kedua sedang memilih karakter, pemain kedua belum terhubung dengan *server* Photon, karakter pemain kedua belum terinstansiasi, atau tidak ada pemain kedua.



**Gambar 5.5 Antarmuka Battle Menu – dua karakter telah terinstansiasi**

Gambar 5.5 menunjukkan bahwa kedua karakter telah terinstansiasi dan permainan siap untuk dimulai.

#### **5.2.2.2 PF-03: Pengujian Perubahan HP dan MP Karakter pada Battle Menu**

Perubahan HP dan MP karakter ditunjukkan dengan *bar* berwarna hitam transparan pada pojok kanan atas layar. Panjang *bar* Hp dan MP akan berubah sesuai dengan jumlah HP dan MP karakter. Perubahan panjang HP dan MP karakter ditunjukkan pada Gambar 5.6.



**Gambar 5.6 Perubahan HP dan MP**

HP karakter akan berkurang jika terkena serangan dari lawan sedangkan MP akan bertambah jika serangan mengenai karakter lawan atau jika terkena karakter lawan dan MP akan berkurang jika pemain menggunakan serangan kuat (*Power Attack*) atau serangan istimewa (*Special Attack*).

Notifikasi (*popup*) bahwa HP salah satu karakter telah habis dan permainan telah usai ditunjukkan pada Gambar 5.7.



**Gambar 5.7 Notifikasi (*popup*) bahwa permainan telah usai**

Gambar 5.6 juga menunjukkan bahwa pergerakan karakter menggunakan tombol dan rotasi karakter menggunakan *accelerometer* telah bekerja sesuai dengan yang diharapkan.

### **5.2.2.3 Hasil Pengujian Fungsional**

Subbab ini berisi tentang hasil pengujian fungsionalitas yang sudah dilakukan berdasarkan pada PF01, PF02, dan PF03. Tiga pengujian yang telah dilakukan menunjukkan bahwa semua fungsionalitas permainan berjalan dengan baik dan sesuai dengan apa yang diharapkan serta sesuai dengan skenario dan alur yang telah dibuat pada perancangan. Rekapitulasi hasil pengujian fungsionalitas ditunjukkan pada Tabel 5.6.

**Tabel 5.6 Rekapitulasi hasil pengujian fungsional**

No	Kode Pengujian	Hasil Pengujian
1.	PF01	Berhasil
2.	PF02	Berhasil
3.	PF03	Berhasil

### **5.3 Pengujian Pengguna**

Pengujian pada perangkat lunak yang dibangun tidak hanya dilakukan pada fungsionalitas yang dimiliki, tetapi juga pada pengguna untuk percobaan secara langsung. Pengujian ini berfungsi sebagai pengujian subjektif yang bertujuan untuk mengetahui tingkat keberhasilan aplikasi yang dibangun dari sisi pengguna. Hal ini dapat dicapai dengan meminta penilaian dan tanggapan dari pengguna terhadap sejumlah aspek perangkat lunak yang ada.

#### **5.3.1 Skenario Uji Coba Pengguna**

Dalam melakukan pengujian perangkat lunak, penguji diminta untuk mencoba menggunakan perangkat lunak yang bersangkutan untuk mencoba semua fungsionalitas dan fitur yang tersedia

Pengujian aplikasi oleh pengguna dilakukan dengan sebelumnya memberikan informasi seputar aplikasi, kegunaan, dan fitur-fitur yang dimiliki. Setelah informasi tersampaikan, pengguna kemudian diarahkan untuk langsung mencoba aplikasi dengan spesifikasi lingkungan yang sama dengan yang telah diuraikan pada uji coba fungsionalitas.

Jumlah pengguna yang terlibat dalam pengujian perangkat lunak sebanyak tiga orang. Dalam melakukan

pengujian, pengguna melakukan percobaan lebih dari satu kali penggunaan untuk masing-masing pengguna.

Dalam memberikan penilaian dan tanggapan, penguji diberikan formulir pengujian perangkat lunak. Formulir pengujian perangkat lunak ini memiliki beberapa aspek penilaian dan pada bagian akhir terdapat saran untuk perbaikan fitur.

### 5.3.2 Daftar Penguji Perangkat Lunak

Pada subbab ini ditunjukkan daftar pengguna yang bertindak sebagai penguji coba aplikasi yang dibangun. Daftar nama penguji aplikasi ini ditunjukkan pada Tabel 5.7.

**Tabel 5.7 Daftar Nama Penguji Coba Aplikasi**

<b>Nomor</b>	<b>Nama</b>	<b>Pekerjaan</b>
1	Yuan Akbarsyah Pandunegoro	Mahasiswa Teknik Informatika ITS
2	Sandhi Ading Wasana	Mahasiswa Teknik Informatika ITS
3	Rizqi Hidayatullah	Mahasiswa Teknik Informatika ITS
4	Wahyu Widodo	Mahasiswa Teknik Informatika ITS
5	Bagus Arief Setiawan	Mahasiswa Teknik Informatika ITS
6	Stieven Wirakasa	Mahasiswa Teknik Informatika ITS

### 5.3.3 Hasil Uji Coba Pengguna

Uji coba yang dilakukan terhadap beberapa pengguna memiliki beberapa aspek yang dipisahkan berdasarkan antarmuka

dan performa sistem. Sistem penilaian didasarkan pada skala penghitungan satu sampai empat di mana skala satu menunjukkan nilai terendah dan skala empat menunjukkan skala tertinggi. Skala dan kategori penilaian ditunjukkan pada Tabel 5.8. Penilaian akhir dilakukan dengan menghitung berapa banyak penguji yang memilih suatu skala tertentu kemudian dihitung nilai rata-rata dari nilai-nilai tersebut. Hasil uji coba ditunjukkan pada subbab-subbab berikut secara lengkap dengan disertai tabel.

**Tabel 5.8 Skala dan kategori penilaian**

Nilai	Keterangan
1	Kurang
2	Cukup
3	Baik
4	Sangat Baik

### 5.3.3.1 Hasil Penilaian Antarmuka

Penilaian antarmuka difokuskan pada penilaian pengguna terhadap kemudahan penggunaan antarmuka dan fitur-fitur lain yang diperlukan. Hasil penilaian pengguna terhadap antarmuka aplikasi ditunjukkan pada Tabel 5.9.

**Tabel 5.9 Penilaian Antarmuka**

No.	Antarmuka	Penilaian				Rata-Rata
		1	2	3	4	
1	Kemudahan Penggunaan	0	0	2	4	3,7



2	Kelengkapan Menu	0	0	1	5	3,8
3	Keindahan Tampilan	0	0	2	4	3,7
4	Kesesuaian <i>audio</i>	0	0	1	5	3,8
5	Kecepatan Pemilihan Menu/Fitur	0	1	2	3	3,3
6	Kesesuaian tema	0	0	3	3	3,5
7	Ketertarikan Bermain	0	1	3	2	3,1
<b>Nilai Akhir</b>						<b>3,557</b>

### 5.3.3.2 Hasil Penilaian Performa Sistem

Penilaian performa sistem difokuskan pada penilaian pengguna terhadap kemampuan aplikasi dalam hal performa atas interaksi pengguna. Penilaian ini juga ditujukan untuk mendapatkan tingkat kecepatan dan kelancaran sistem atas interaksi yang dibuat oleh pengguna. Hasil penilaian performa sistem ditunjukkan pada Tabel 5.10.

**Tabel 5.10 Penilaian Performa Sistem**

No.	Performa Sistem	Penilaian				Rata-Rata
		1	2	3	4	
1	Performa atau kinerja permainan	0	0	2	4	3,7
2	Performa atau kinerja fitur AR	0	0	3	3	3,5
3	Performa atau kinerja fitur <i>multiplayer</i>	0	0	1	5	3,8
4	Performa atau kinerja model dan animasi 3D	0	0	1	5	3,8
5	Nilai dinamis permainan	0	0	3	3	3,5

6	Ketepatan dalam hal kontrol karakter dengan tombol dan <i>accelerometer</i>	0	0	3	3	3,5
7	Tampilan karakter dan kelancaran animasi	0	0	1	5	3,8
8	Kesesuaian animasi dengan sistem permainan	0	0	2	4	3,7
9	Kesesuaian interaksi dengan hasil yang seharusnya	0	0	2	4	3,7
<b>Nilai Akhir</b>						<b>3,667</b>

### 5.3.4 Hasil Pengujian Pengguna

Evaluasi pengujian pengguna dilakukan dengan menampilkan data rekapitulasi perangkat lunak yang telah dipaparkan. Dalam hal ini, rekapitulasi disusun dalam bentuk tabel yang ditunjukkan pada Tabel 5.11. Dari data tersebut, diketahui bahwa aplikasi telah memenuhi unsur yang seharusnya dimana nilai prosentase adalah lebih dari 80%.

**Tabel 5.11 Rekapitulasi Hasil Uji Coba Pengguna**

No.	Nama Pengujian		Rata-Rata	Nilai	Nilai (%)
1	Penilaian Antarmuka	Kemudahan Penggunaan	3,7	3,557	88.925
		Kelengkapan Menu pada Permainan	3,8		
		Keindahan Tampilan	3,7		
		Kesesuaian <i>audio</i>	3,8		
		Kecepatan Pemilihan Menu/Fitur	3,3		

		Kesesuaian tema pada Permainan	3,5		
		Ketertarikan Bermain	3.1		
2	Performa Sistem permainan	Performa atau kinerja pada permainan	3,7	3,667	91.675
		Performa atau kinerja fitur AR	3,5		
		Performa atau kinerja fitur <i>multiplayer</i>	3,8		
		Performa atau kinerja model dan animasi 3D	3,8		
		Nilai dinamis permainan	3,5		
		Ketepatan dalam hal kontrol karakter	3,5		
		Tampilan karakter dan kelancaran animasi	3,8		
		Kesesuaian animasi dengan sistem permainan	3,7		
		Kesesuaian interaksi dengan hasil yang seharusnya	3,7		
		Kecocokan permainan untuk target pengguna	3,5		

		Nilai inovasi permainan	3,7		
		Kepraktisan permainan	3,5		

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

#### **6.1. Kesimpulan**

Dalam proses pengerjaan Tugas Akhir ini, mulai dari tahap analisis, desain, implementasi, hingga pengujian didapatkan kesimpulan sebagai berikut:

1. Berdasarkan uji antarmuka, diketahui bahwa antarmuka aplikasi mudah digunakan, memiliki tampilan yang lengkap dan indah, dan menggunakan *audio* yang sesuai.
2. Berdasarkan uji performa, diketahui bahwa aplikasi telah berhasil mengimplementasikan *game* yang ber-*genre fighting* dan mengakomodasi teknologi AR dengan kontrol yang sesuai. Animasi karakter yang berbasis 3D juga berjalan dengan lancar.
3. Aplikasi telah berhasil mengimplementasikan fitur *multiplayer* secara *online* sehingga menjadikan permainan semakin dinamis. Berdasarkan uraian yang telah disampaikan, dapat disimpulkan semua fungsionalitas permainan berfungsi dengan baik.

#### **6.2. Saran**

Berikut adalah beberapa saran untuk pengembangan sistem di masa yang akan datang berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.

1. Kontrol karakter menggunakan cara yang lebih mudah sehingga pemain tidak merasa repot dengan kontrol tersebut.

2. Arena sebaiknya tidak terlalu besar agar ruang gerak karakter tidak terlalu lebar.
3. Jarak serangan karakter ditambah agar serangan dapat mengenai karakter lawan dengan lebih mudah.

## DAFTAR PUSTAKA

- [1] Nicolescu, A. (2014, April 28). *Black box techniques*. Retrieved from <https://www.qualitance.com/blog/black-box-techniques> [Diakses pada tanggal 9 September 2015]
- [2] Kim, J., & Ae-Hyun. (n.d.). *Design and Development of Unity3D Game Engine-Based Smart SNG (Social Network Game)*.
- [3] *Vuforia Augmented Reality SDK*. (n.d.). from [https://en.wikipedia.org/wiki/Vuforia\\_Augmented\\_Reality\\_SDK](https://en.wikipedia.org/wiki/Vuforia_Augmented_Reality_SDK) [Diakses pada tanggal 27 Desember 2015]
- [4] *Photon Unity Networking*. (n.d.). from <http://doc-api.photonengine.com/en/pun/current/> [Diakses pada tanggal 29 Desember 2015]
- [5] *Android (sistem operasi)*. (n.d.). from [https://id.wikipedia.org/wiki/Android\\_%28sistem\\_operasi%29](https://id.wikipedia.org/wiki/Android_%28sistem_operasi%29) [Diakses pada tanggal 29 Desember 2015]
- [6] *Fungsi accelerometer & gyroscope pada smartphone*. (n.d.). from <http://www.andromin.com/2015/04/fungsi-accelerometer-gyroscope-pada-android.html> [Diakses pada tanggal 28 Desember 2015]
- [7] *Vuforia Augmented Reality SDK*. (n.d.). from [https://en.wikipedia.org/wiki/Vuforia\\_Augmented\\_Reality\\_SDK](https://en.wikipedia.org/wiki/Vuforia_Augmented_Reality_SDK) [Diakses pada tanggal 27 Desember 2015]
- [8] *Fighting game*. (n.d.). from [http://en.wikipedia.org/wiki/Fighting\\_game](http://en.wikipedia.org/wiki/Fighting_game) [Diakses pada tanggal 27 Desember 2015]
- [9] *Blender Reference Manual*. (n.d.). from [https://www.blender.org/manual/getting\\_started/about\\_blender/introduction.html](https://www.blender.org/manual/getting_started/about_blender/introduction.html) [Diakses pada tanggal 29 Desember 2015]

## BIODATA PENULIS



Penulis dilahirkan di Sidoarjo pada tanggal 21 Oktober 1994, merupakan anak pertama dari dua bersaudara. Penulis telah menempuh pendidikan formal yaitu TK Asy-Syuhada' (1998-2000), MI Ma'arif Pagerwojo (2000-2006), SMP NEGERI 1 Sidoarjo (2006-2009), Madrasah Aliyah Amanatul Ummah Program Akselerasi (2009-2011), dan S1 Jurusan Teknik Informatika dengan rumpun mata kuliah Interaksi, Grafika, dan Seni (2011-2016).

Selama menjadi mahasiswa, penulis pernah menjadi juara 1 Lomba Panco TC Cup 2014 dan pembicara pada ICTS 2015 serta menjadi koordinator berbagai macam acara kemahasiswaan. Penulis juga aktif dalam organisasi kemahasiswaan CSSMoRA ITS. Penulis memiliki hobi olahraga yaitu *jogging*, *fitness*, *calisthenics*, *body building*, dan *entrepreneur*. Penulis pernah berkecimpung dalam bidang non-akademik, misalnya menjadi *partner* di *gym*. Penulis dapat dihubungi melalui surel [alfian.maulanaazhari@hotmail.com](mailto:alfian.maulanaazhari@hotmail.com).